commodore Magazine

AÑO II - Núm. 11 - Enero 1985 - 250 Ptas.

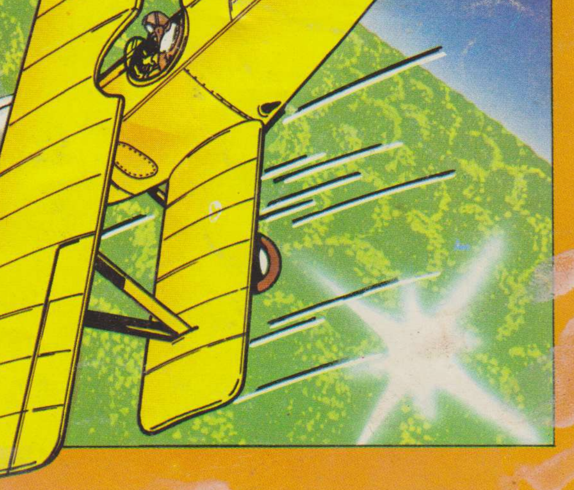
REVISTA INDEPENDIENTE PARA USUARIOS

Fusas y Corcheas en el C-64



Juego de aventuras: SUPERVIVENCIA



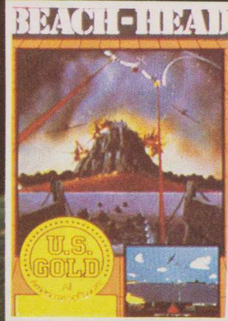




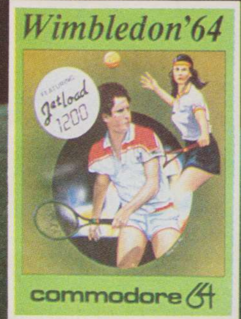
microbyte

PROGRAMAS ORIGINALES DE IMPORTACION PARA commodore 64

TODOS ESTOS PROGRAMAS HAN ESTADO SITUADOS ENTRE LOS CINCO PRIMEROS PUESTOS DE LAS LISTAS DE SUPERVENTAS BRITANICAS



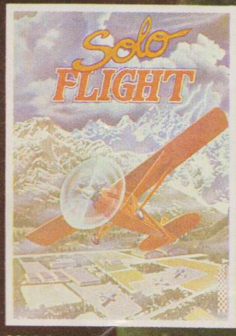
BEACH HEAD. El juego más popular de América con los más sorprendentes efectos de animación (COMP. ANSWER). Los mejores efectos de sonido y gráficos para este computador (U.S. BILLBOARD). Múltiples pantallas, fantásticos efectos tridimensionales en un escenario de acción bélica. Un best-seller (GAMES). 2.300 pts.



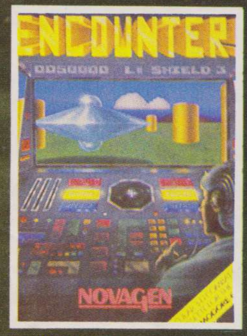
WIMBLEDON 64. Uno de los mejores juegos de simulación de deportes concebido para C-64 (C. & V. GAMES). Horas de entretenimiento para los amantes del tenis y admiradores de McEnroe, Connors y Borg (POP. COMP. WKLY). Los gráficos, la movilidad y el control de las jugadas es fántastico (YOUR 64). 2.300 pts.



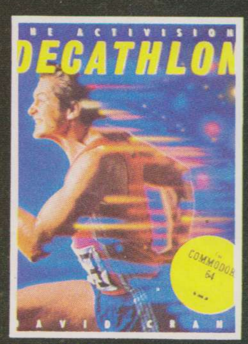
MANIC MINER. Verdaderamente supera a la versión líder de ventas para Spectrum (COM. USGR.). Pantallas y pantallas con todo tipo de gráficos, colores y efectos musicales (HOME. COMP. WKLY). Superior a "Miner 2049". Sin más (CRASH). 1.800 pts.



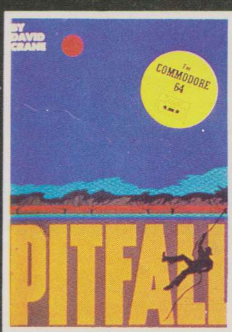
SOLO FLIGHT. Incomparable, sólo necesita pista y alas (COMPUTE). Constantes efectos tridimensionales, perfecto scroll y auténticos efectos sonoros (RUN). 21 aeropuertos diferentes, idénticos instrumentos de bordo, emergencias, V.O.R., una perfecta simulación de vuelo (COM. USER). 3.900 pts.



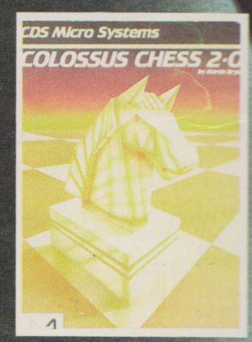
ENCOUNTER. Los gráficos son perfectos, el efecto tridimensional soberbio. Este juego es un "primera clase" (WICH MICRO). ENCOUNTER es el juego de salón más completo escrito para un ordenador personal (GAMES). Nuestra puntuación: un 10 (TPUG). 2.300 pts.



DECATHLON. La animación y los gráficos de los atletas es soberbia (P.C. GAMES). Decathlon es una bella simulación de los 10 eventos de la popular prueba olímpica (POPULAR COMPUTING). Uno de los programas más adictivos y espectaculares que he conocido (CRASH). 2.800 pts.



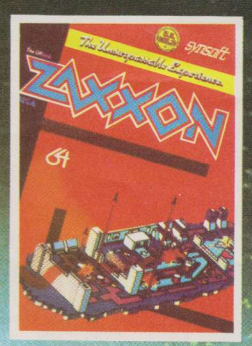
PITFALL. Uno de los juegos para "Atari" más vendidos en U.S.A. ahora en COMMODORE (WHAT MICRO). Caimanes, serpientes, escorpiones, una selva en tu 64 (GAMES COMPUTING). ¿Quién ha sido capaz de encontrar los 32 tesoros escondidos en menos de 20 minutos? ... (ELECTRONICS & GAMES). 2.800 pts.



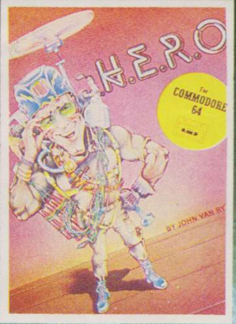
COLOSSUS CHESS 2º. Este juego de inteligencia es indiscutible en vuestra biblioteca (GAMES). El juego mejor logrado en su categoría (POP. COMP. WKLY). Brillante sin duda 4 estrellas (COMP. & VIDEO GAMES). 2.800 pts.



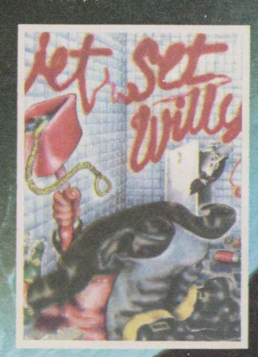
ASTRO CHASE. 34 pantallas con caricaturas animadas en tres dimensiones (GAMES). Juego de fantasía y ciencia ficción del año 1984 (ELECTRONIC GAMES). La mayor innovación en videojuegos (THE WHIZ KID). Este programa batirá todos los records de superventas (RUN). 2.300 pts.



ZAXXON. Increíbles gráficos en tres dimensiones con magníficos scrolls diagonales y un supersónido son los ingredientes de este best seller (POP. COMP.). Todas las versiones que han aparecido (Apple, Atari, Coleco, etc.) han sido números uno, sin duda ésta lo será también (U.S. BILLBOARD). 2.300 pts.



H.E.R.O. Este programa es el más original de la nueva ola de programas sobre minas y laberintos. El intrépido HERO deberá rescatar a los mineros con su equipo especial de salvamento (ELECTRONIC GAMES). Extraordinarios gráficos y ambientación (HOME COMP. WKLY). 2.800 pts.



JET SET WILLY. Excelente, altamente recomendado (GAMES). Color y sonido excelente, gráficos perfectos: nuestra puntuación: un 10 (COMP. & VIDEO GAMES). 1.800 pts.

COMMODORE-64

TRON	1.700
FROGGER (Joystick)	1.700
KONG	1.700
SCREEN GRAPHICS	1.800
ENSAMBLADOR	1.900

VIC-20

JOY STICK PAINTER	3, 8, 16K	1.600
PIPER	3, 8, 16K	1.700
TRON	Std.	1.600
INVASION	Std.	1.600
PHANTON	3, 8, 16K	1.600
PARATROPERS	Std.	1.600

Envíenos a **MICROBYTE** San Gerardo, 59 MADRID-35

	Juego	Precio	TOTAL
Nombre			
Apellidos			
Dirección			
Población			
D.P. Teléfono			
Incluyo talón nominativo ☐	PRECIO TOTAL PESETAS		
Contra-Reembolso ☐			
C.M. ENVIOS GRATIS	Pedidos por Teléfono 91-6565002		

commodore
Magazine

Sumario

Año 1
Núm. 11

Director:
Alejandro Diges
Coordinador editorial:
J. A. Sanz
Redacción:
Roberto Menéndez
Aníbal Pardo
Teresa Aranda
Gumersindo García
Fernando García
Diseño:
Ricardo Segura

Editada por
Publinformática
Presidente:
Fernando Bolín
Director Editorial:
Norberto Gallego

Administración:
INFODIS S.A.
Gerente de Circulación y Ventas:
Luis Carrero
Producción:
Miguel Onieva
Director de Marketing:
Antonio González
Servicio al cliente:
Julia González - Tel.: 733 79 69
Administración:
Miguel Atance, Antonio Torres
Jefe de Publicidad:
María José Martín

Dirección y Redacción
C/Bravo Murillo, 377 - 5º A
Tel.: 733 74 13
28020 - MADRID
Publicidad y Administración:
C/ Bravo Murillo, 377 - 3º E
Tel. 733 96 62/96

Publicidad en Madrid:
Fernando Hernando
Publicidad en Barcelona:
María del Carmen Ríos
y Jorge González
Pelayo, 12.
Tel.: (93) 301 47 00 ext. 27 y 28
08001 - BARCELONA
Depósito Legal: M-6622-1984
Distribuye: S.G.E.L.
Avda. Valdelaparra, s/n.
Alcobendas. Madrid

Fotocomposición: Consulgraf
C/Nicolás Morales, 34. Madrid
Fotomecánica: Karmat
C/Pantoja, 10. Madrid
Imprime: Novograph, S.A.
Carretera de Irún, Km. 12,450
Madrid.
Solicitado control O.J.D.

Esta publicación es miembro de la Asociación de Revistas de la Información ARI, asociada a la Federación Internacional de Prensa Periódica FIPP.

ari

ROGAMOS DIRIJAN TODA LA CORRESPONDENCIA RELACIONADA CON SUSCRIPCIONES A:
COMMODORE MAGAZINE
EDISA: Tel. 415 97 12
C/ López de Hoyos, 141-5.
28002-MADRID
PARA TODOS LOS PAGOS RESEÑAR SOLAMENTE COMMODORE MAGAZINE

ROGAMOS QUE PARA LA COMPRA DE EJEMPLARES ATRASADOS SE DIRIJAN A LA PROPIA EDITORIAL
commodore Magazine
C/ Bravo Murillo, 377-5. A
Tel. 733 74 13
28020-MADRID

Esta revista no mantiene relación de dependencia de ningún tipo con respecto de los fabricantes de ordenadores Commodore Business Machines ni de sus representantes.

Editorial

La reciente introducción del ordenador C-16 hace pensar que *Commodore Business Machines* irá consolidándose cada vez más en el difícil y concurrido mercado del micro de bajo precio. En estas mismas páginas, escribíamos sobre la presentación de los nuevos modelos, hace pocos meses.

Por otro lado, la multinacional americana no parece tener demasiada prisa por lanzar definitivamente los modelos de la gama orientada a los profesionales. Nos referimos concretamente a las anunciadas máquinas: compatible con el *PC* de *IBM* y la basada en el microprocesador Z-8000. Recordemos el impacto que produjo *CBM* cuando compró la firma canadiense Hyperion, fabricante de un compatible *IBM* con pantalla de 9" incorporada y teclado separable. Aunque en la Feria de Hannover vimos el modelo que se comercializaba originalmente, al que solamente le habían añadido etiqueta con el logotipo de *Commodore,* esta última se apresuró a comentar un nuevo diseño de la carcasa y diversas mejoras respecto al modelo original. Hasta el momento no parece haber prisas por lanzarlo.

Incluso la posibilidad de fabricar el chip microprocesador 8086, bajo licencia por parte de *MOS Technologies,* subsidiaria de *CBM,* hizo pensar en una fuerte entrada de esta firma en el mercado de los compatibles. De alguna manera, el *C-16* consolidaría a *Commodore* en el segmento de mercado y precio que en su día abrió el legendario *Vic-20.*

En este número hemos abierto una línea de programas orientados a quienes gustan de los juegos de aventuras. Lamentablemente Vic-20 no es tan apto para las aventuras, que necesitan de un programa largo, por la limitación de la memoria, pero buscaremos alguno con interés suficiente para el lector. La manera más familiar para los músicos de producir melodías con el *64* también es abordada en este número, con la revisión del conjunto teclado + software, Music 64.



Códigos de control para el VIC-20 y el C-64

COMO SE VE	COMO SE TECLEA	EFECTO CONSEGUIDO
COLORES DEL VIC-20 Y DEL 64		
	CTRL+1	NEGRO
	CTRL+2	BLANCO
	CTRL+3	ROJO
	CTRL+4	CIAN
	CTRL+5	PURPURA
	CTRL+6	VERDE
	CTRL+7	AZUL
	CTRL+8	AMARILLO
COLORES DEL 64 SOLAMENTE		
	CBM+1	NARANJA
	CBM+2	MARRON
	CBM+3	ROSA
	CBM+4	GRIS OSCURO
	CBM+5	GRIS MEDIO
	CBM+6	VERDE CLARO
	CBM+7	AZUL CLARO
	CBM+8	GRIS CLARO
CODIGOS DE CURSOR Y CONTROL		
	HOME	CURSOR A CASA
	SHIFT+HOME	LIMPIA PANTALLA
	CRSR	CURSOR DERECHA
	SHIFT+CRSR	CURSOR IZQUIERDA
	CRSR	CURSOR ABAJO
	SHIFT+CRSR	CURSOR ARRIBA
	CTRL+9	CARACTER INVERSO
	CTRL+0	CARACTER NORMAL
	DEL	BORRAR
	SHIFT+DEL	INSERTAR
TECLAS DE FUNCION		
	F1	
	F2=SHIFT+F1	
	F3	
	F4=SHIFT+F3	
	F5	
	F6=SHIFT+F5	
	F7	
	F8=SHIFT.T+F7	

P.: Me gustaría que me ayudáseis a resolver un problema que tengo con mi **Vic-2Ø.**

Resulta que estoy haciendo un pequeño programa de aplicaciones para mi trabajo y la cuestión es que quiero introducir datos con la cláusula INPUT y me sale bien, pero lo que no quiero es que me salga el interrogante, ya que me queda muy feo y me estropea el programa en general.

J. F.º Ferrando

R.: Hace poco tiempo publicamos la respuesta para el mismo problema, planteada para un **C-64.** En el caso del **Vic-2Ø** es exactamente análoga. Partimos del conocimiento de que el teclado es tratado por el **Vic 2Ø**, como el dispositivo que tiene asignado el número Ø. Esta posibilidad la podemos tratar con fines de Entrada/Salida. Es decir, el teclado actúa como si fuera un dispositivo de entrada de datos en el ordenador. Así la utilización de INPUT# introduce lo datos sin que tenga porqué aparecer el simbolo "?".

La siguiente rutina breve te dará la respuesta apetecida:

Existe otra forma de conseguir lo mismo. Consiste en poner a 1 la dirección de la memoria 19, antes de incluir la sentencia con INPUT, desapareciendo la "?". Una vez introducido el dato, debemos devolver su contenido, poniendo a Ø esta dirección:

Esperamos que la respuesta sea satisfactoria para tu programa.

P.: Tengo un **C-64** y he leído muchas veces sobre cómo programar las teclas de función. La mayoría de las veces se utiliza la sentencia GET. ¿Existe otra manera de detectar cuando es presionada una tecla de función?; ¿existe alguna dirección que pueda PEEKear?

R.: Sí, la dirección de memoria 197, tanto en el **Vic-20** como en el **C-64,** indica la tecla que estamos pulsando en un momento dado. Utilice la siguiente línea de programa:

```
10 PRINT PEEK (197) : GOTO 10
```

Mientras está funcionando este programa presione una tecla de función. Como puede ver, cada vez que activa una tecla el valor que aparece en la pantalla cambia. Si no presionamos ninguna, el valor es 64. El valor correspodiente a cada una de las teclas es el siguiente para cada ordenador:

TECLA	Vic-20	C-64
F1	39	4
F3	47	5
F5	55	6
F7	63	3

El problema de este método es que el valor mostrado es el mismo indepedientemente de que se presione Shift o no. Para solventarlo podemos recurrir a PEEKear otra dirección de la memoria, la 653, que indica si ha sido presionada SHIFT, CTRL o la tecla **Commodore.** El bit 0 se ponde a 1 con la presión de SHIFT, la tecla **Commodore** afecta a bit 1 y CTRL SHIFT, la tecla **Commodore** afecta al bit y CTRL al 2.

P.: Contesto a su respuesta a Pedro J. Boj de Alicante en el n.º 8.

Yo también soy un resignado propietario de **PET 2001**-32K + 2 drive **Cumputhink + Centronics** 781.

Existimos y probablemente seamos bastantes, pero creo que los demás como yo estamos acostumbrados a que no cuenten con nosotros.

En mi caso, pienso no cambiar de equipo, mientras me lo arreglen. Ya que es operativo para los programas técnicos que he desarrollado.

De todas maneras miro con un poco de envidia lo que se publica y no puedo utilizar. Tengo la esperanza de que alguna vez se acuerden de nosotros. Animen a los PETistas a salir a la luz, aunque sea por mantener nuestro interés por la revista.

J.L. Resines. Las Palmas de Gran Canaria.

R.: Nos alegramos mucho de recibir esta carta. Con ella aprovechamos para hacer lo que nos sugiere: «animar a los PETistas», pero es que están de lo más callado.

En **Commodore Magazine** estamos dispuestos a recibir las colaboraciones de cuantos hayan desarrollado algo para los legendarios **PET.** Esta invitación es nuestra única posibilidad de publicar algo, pues carecemos del equipo, necesidad básica para poder hacer algo. Incluso trataríamos de hacer el esfuerzo que implica hacer determinado tipo de adaptaciones. ¿Por qué no publicar uno de esos programas técnicos que nos menciona? Esperamos que los poseedores del **PET** al menos nos escriban para hacernos conocer su existencia e inquietudes informáticas.

P.: Por favor, desearía recibir información sobre todos los juegos que existen actualmente para el **Commodore 64.**

Si es posible mándenme un folleto o catálogo.

También los precios correspondientes a cada juego.

J.M.ª Lladó. Granada.

R.: Tu carta tiene un contenido similar al de muchas otras que recibimos. Respondiéndote esperamos responderos a todos.

Commodore Magazine es una revista encuadrada en el entorno de una editorial. Nosotros no vendemos ordenadores, periféricos o software, únicamente revistas.

De todas las maneras, dado el interés que es evidente existe por conocer la realidad de mercado, **Commodore Magazine** está realizando una guía de *software*, que se incluirá en las páginas de la revista en un próximo número.

De cualquier manera, lamentamos no poder facilitaros la información comercial que nos solicitáis.

Music-64



Este es el formidable aspecto que ofrece el teclado Music-64.

Es un hecho conocido que el **Commodore 64** incorpora en su circuitería interior **chips** exclusivos de aplicación específica, que liberan al microprocesador de realizar determinadas tareas. Este es el caso del *chip* **SID**, un verdadero sintetizador de sonidos, cuyas posibilidades son casi ilimitadas en las manos de un programador experto. Sin embargo, no sirve de mucho desarrollar un prodigioso programa de síntesis musical si luego ha de ejecutarse desde el teclado del ordenador. No parece que muchos músicos sientan pasión por la taquimecanografía, aunque esta vez las teclas produzcan notas musicales en vez de letras y números. Tal vez lo que necesita el **64** es un verdadero teclado similar al de un piano.

El **Music 64** es un teclado de cuatro octavas destinado únicamente a trabajar con el **Commodore 64.** El aspecto físico del conjunto recuerda a los órganos electrónicos clásicos. La carcasa de plástico que lo alberga es del mismo tipo y color que la empleada por el fabricante del ordenador. Por la parte posterior del teclado sale un cable plano, que en su otro extremo dispone de un cartucho del mismo tipo empleado para los programas en ROM, y se conecta en la ranura destinada a enchufarlos en el **64.** En realidad no queda mucho más que describir del exterior.

Para que el teclado nos ofrezca una verdadera utilidad práctica es necesario el *software* que asegure su funcionamiento. Existe una versión en *cassette* y otra en *diskette.* Durante nuestro examen dispusimos de la segunda versión. Aun no siendo virtuosos de la ejecución musical nos encontrábamos ansiosos por ponerle la mano, mejor dicho los dedos, encima a este atractivo conjunto.

Una vez conectado el cartucho del

teclado, presionamos el conmutador de alimentación del **64.** Tras el familiar READY decidimos leer el directorio de los ficheros de programas contenidos en el *diskette,* y aparecen tres nombres en la pantalla: «Mono 64», «Poly 64» y «Gestione».

Seguimos leyendo la hoja que constituye el manual y la primera referencia se hace al Mono 64, explicando que se trata del sintetizador monofónico. Pues bien, cargamos este programa en la memoria. Hagamos un inciso para aclarar que la hoja/manual no es demasiado clara, pues contiene giros raros hechos durante la traducción, pero esto no desmerece en absoluto a la magnífica calidad del equipo.

Una vez que hemos cargado el programa y este se ejecuta, en la pantalla aparece el título del mismo, inmediatamente debajo un diagrama alegórico al teclado y la indicación, todo en inglés, de que debemos esperar un instante atentos a la pantalla de nuestro televisor o monitor. Inmediatamente después se vuelve a encender el led, que indica la operación de la unidad de *diskettes,* y comienza a cargarse el programa «gestione», que convive en la memoria con el anterior.

Cuando termina la carga del *software,* el ordenador visualiza la primera pantalla que nos permite comenzar a trabajar. Es realmente el panel frontal de un sintetizador compacto.

Seis rectángulos forman el conjunto, cada uno con su propia misión. El situado en la esquina superior izquierda (ver foto) se encarga de la envolvente, por medio del control de los cuatro parámetros, que establecen las magnitudes de Ataque, Decaimiento, Sostenimiento y Relajación. A su derecha está el rectángulo que controla el filtro, con otros cuatro parámetros: Modo 1, p, Corte, Modulación ADSR (con el generador de envolvente) y Resonancia. El tercero de la fila superior se encarga del control de los tres parámetros del oscilador de muy baja frecuencia (L.F.O.), que son Velocidad, Modulación del filtro (V.C.F. = Voltage Controled Filter) y Modulación del oscilador.



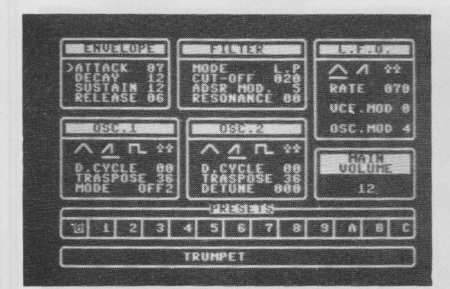
Detalle de la conexión del teclado al ordenador.

Debajo se localizan los dos rectángulos que afectan a los dos osciladores, que son en realidad las voces o canales 1 y 3 del **SID**, pudiendo elegir entre las cuatro conocidas formas de onda: Ruido, Diente de sierra, Impulsos y Triangular, aparte de estar previstos otros ajustes tales como la duración del tiempo activo del ciclo del impulso. Por último, el rectángulo situado en la parte inferior derecha controla el nivel del volumen de la salida de audio entre 0 y 15, como se hace normalmente en el **SID**.

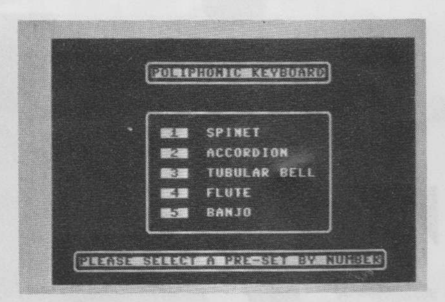
Probablemente con esta breve explicación de lo que hay, parezca difícil ejercer el control sobre el panel de la pantalla, pero nada más lejos de la realidad. Las cuatro teclas de función facilitan enormemente la labor. F1 pasa al parámetro siguiente cuando es presionada. Cuando está en el último parámetro de un rectángulo pasa al primero del siguiente. F2 hace lo mismo, pero en sentido inverso. Cabe aclarar que las teclas gozan de la propiedad de la autorrepetición, es decir, el ordenador interpreta que estamos presionando repetidamente la tecla y el acceso a los parámetros lejanos es más rápido. Por su parte F5 sirve para aumentar el valor del parámetro sobre el que estamos situados. Así podemos hacer que sea más largo el tiempo de subida del período de ataque o elevar el volumen general o hacer que el filtro trabaje en resonancia, etc. La tecla F7 hace lo mismo, pero disminuyendo el valor en lugar de elevarlo.

Paralelamente, en la parte más baja de la pantalla existe una hilera de trece cuadrados que contienen cada uno un número del cero al nueve y las letras A, B y C. Presionando la tecla correspondiente a cada uno de esos números y letras podemos acceder a la síntesis de un instrumento musical diferente. Al mismo tiempo se alteran los valores de los parámetros del panel de acuerdo con el instrumento en cuestión. Para saber en qué momento estamos, el cuadrado correspondiente a nuestra elección visualiza el número o letra en inverso y debajo aparece escrito el nombre del instrumento, que puede ser: Trompeta, Metal, Clarinete, Campanas, Flauta, Guitarra, Metal con Wha, Cuerda, Piano, Organo eléctrico en versión 1 ó 2, Acordeón y Sonidos aleatorios.

Si por cualquier circunstancia de-



Pantalla para ajustar parámetros en modo monofónico (trompeta).



Menú de instrumentos utilizables en modo polifónico.



Pantalla de ajuste de parámetros en modo polifónico (acordeón).

seamos alterar algunos de los parámetros, o todos, simplemente con presionar la tecla de función adecuada a nuestro deseo recobramos el control sobre el panel y la leyenda del instrumento es alterada por otra que dice «Modify Mode», o lo que es igual: trabajamos en el modo que permite modificar los valores. La tecla CLR nos permite recuperar los valores del instrumento que habíamos elegido previamente.

La segunda opción es la del teclado polifónico, que se carga desde el *diskette* llamándolo por su nombre, **Poly 64**. En este caso aparece igualmente un dibujo de teclado de piano y el consabido mensaje que incita a la calma mientras se carga el resto del programa. Con el sintetizador polifónico no tenemos representación de un panel en la pantalla. Podemos comenzar eligiendo uno entre cinco instrumentos: Espineta, Acordeón, Campanas tubulares, Flauta o Banjo. Eligiendo una de las cinco opciones de este menú aparece otra pantalla en la que ocho leyendas representan a otros tantos parámetros, que pueden ser accedidos mediante las cuatro teclas de función F1 a F7 y la tecla SHIFT. Al lado de la leyenda identificadora de cada parámetro aparece un número que cuantifica el valor de cada característica del sonido. Presionando la tecla exponente (flecha hacia arriba) el ordenador da permiso al teclado para comenzar la ejecución, a su lado aparece la leyenda «play». La tecla M nos devuelve al menú anterior si lo deseamos.

Con las teclas de función podemos controlar los siguientes parámetros, por este orden: Ataque, Decaimiento, Sostenimiento, Relajación, Volumen global, Frecuencia, Forma de onda y Ciclo activo, en el modo de oscilador con impulsos.

Cuando alteramos los valores preestablecidos por el ordenador, el número correspondiente a ese parámetro cambia a color rojo, para indicarlo. Se pueden hacer tantas modificaciones como se quiera y también podemos recuperar los valores originales presionando la tecla R (para Reset o poner en el estado inicial).

Igual que sucede con el monofónico, el polifónico también dispone de una línea en la parte inferior que nos indica con qué instrumento estamos trabajando o si estamos efectuando modificaciones.

La tarea de revisión de este *hardware* ha sido entretenida y amena. Quien más, quien menos ha hecho sus pinitos tocando alguna melodía sencilla en nuestra redacción, pero los carentes de oído musical se han deleitado simulando efectos especiales con el ruido blanco, el generador de envolvente y el filtro. Los resultados han sido asombrosos en algunos casos.

Programas

SINTETIZADOR 64



El *chip* **SID** del **Commodore 64** es bien conocido como versatil sintetizador de sonidos, pero su principal inconveniente es que hay que programarlo mediante POKEs a sus distintos registros.

Uno de los principales problemas que se encuentran a la hora de programar un juego es encontrar los sonidos adecuados, teniendo que modificar continuamente los registros del **SID** hasta encontrar los más adecuados.

Este programa simplifica la creación de estos sonidos, pues en la pantalla se pueden visualizar todas las características del sonido que se va a crear.

Una vez cargado el programa, por el procedimiento habitual, se utiliza de la siguiente manera:

— Las teclas de función: **F1, F3, F5 y F7,** controlan la magnitud de ADSR.

— Las teclas de movimiento del cursor sirven para seleccionar la nota y la octava del sonido.

— Con las teclas **S** (sierra), **T** (triangular), **R** (ruido) y **P** (impulsos) se elige la forma de onda.

— Elegida la opción de impulsos se introduce la anchura del mismo tecleando un número comprendido entre 0 y 4095, después de haber presionado la tecla **A.**

— La tecla **D** sirve para ajustar la duración de la nota, comprendida también en la rango 0 y 4095, como en el caso anterior.

La tecla de la flecha hacia arriba (exponente) visualiza cuáles son los valores a introducir en los registros con POKE, una vez que llegamos a tener plena satisfacción con el sonido.

— La tecla que lleva impresa la flecha que señala a la izquierda nos permite escuchar el sonido que hemos creado.

El programa va protegido, de tal manera que una vez puesto en marcha no se puede interrumpir, modificar, ni introducir valores fuera del rango.

Es posible introducir este programa junto con el **64 Musical,** publicado en el n.º 3 de **Commodore Magazine,** aunque es preciso efectuar algunas pequeñas modificaciones.

Francisco Acosta

Programas

```
1 REM  ********************
2 REM  * SINTETIZADOR 64 *
3 REM  *                 *
4 REM  *    FCO. ACOSTA    *
5 REM  ********************
10 PRINT"▮▮":POKE53280,9:POKE53281,9:POKE808,225
12 PRINT"▮          SINTETIZADOR 64                    FAG▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮"
15 PRINT"▮▮▮▮▮▮▮UN MOMENTO,POR FAVOR..."
20 S=54272:DIMAF(7,11):DIMBF(7,11)
40 FORE=0TO7:FORN=0TO11:READQ:READW:AF(E,N)=Q:BF(E,N)=W:NEXTN:NEXTE
45 AT=2:DE=2:SO=2:RE=2:CO=55400:DU=250:FO=17:XP=0:YP=0:XA=0:YA=0
46 PA=0:PB=0:TA=0:ED=0:OS=0:ER=0:AD=0:SR=0:NA=1860
47 B$="▮▮||||||||||||||||||||":C$="▮▮|||||||||||||||||||||"
50 FORA=STOS+24:POKEA,0:NEXT
60 POKES+24,15:POKE650,128
450 GOSUB5000
500 GETA$:IFA$=""THEN500
505 IFA$="T"THENFO=17:GOSUB950:POKE56016,3
510 IFA$="S"THENFO=33:GOSUB950:POKE56056,3
515 IFA$="P"THENFO=65:GOSUB950:POKE56136,3
520 IFA$="R"THENFO=129:GOSUB950:POKE56096,3
525 IFA$="▮"THENGOSUB600
530 IFA$="▮"THENGOSUB620
535 IFA$="▮"THENGOSUB640
540 IFA$="▮"THENGOSUB660
545 IFA$="▮"THENGOSUB680
550 IFA$="▮"THENGOSUB700
555 IFA$="▮"THENGOSUB720
560 IFA$="▮"THENGOSUB740
580 IFA$="A"THENGOSUB800
585 IFA$="D"THENGOSUB850
590 IFA$="←"THENGOSUB900
595 IFA$="↑"THENGOSUB1000:GOTO45
597 IFA$="="THEN1200
599 GOTO500
600 AT=AT+1:IFAT=18THENAT=2
605 PRINT"▮▮▮"MID$(B$,1,AT);:PRINTMID$(C$,1,19-AT):TA=AT-2:RETURN
620 DE=DE+1:IFDE=18THENDE=2
625 PRINT"▮▮▮▮▮▮▮"MID$(B$,1,DE);:PRINTMID$(C$,1,19-DE):ED=DE-2:RETURN
640 SO=SO+1:IFSO=18THENSO=2
645 PRINT"▮▮▮▮▮▮▮▮▮▮▮"MID$(B$,1,SO);:PRINTMID$(C$,1,19-SO):OS=SO-2:RETURN
660 RE=RE+1:IFRE=18THENRE=2
665 PRINT"▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮"MID$(B$,1,RE);:PRINTMID$(C$,1,19-RE):ER=RE-2:RETURN
680 YP=YP+1:IFYP=12THENYP=0
685 GOSUB960:RETURN
700 YP=YP-1:IFYP=-1THENYP=11
705 GOSUB960:RETURN
720 XP=XP+2:IFXP=16THENXP=0
725 GOSUB960:RETURN
740 XP=XP-2:IFXP=-2THENXP=14
745 GOSUB960:RETURN
800 CD=1961:POKE56216,3:GOSUB975:POKE56216,8
805 PA=INT(NT/256):PB=INT(NT-(PA*256)):RETURN
850 CD=1977:POKE56240,3:GOSUB975:POKE56240,8:DU=NT:RETURN
900 POKE55960,3:AD=INT((TA*16)+ED):SR=INT((OS*16)+ER)
920 POKES+1,AF(INT(XP/2),YP)::POKES,BF(INT(XP/2),YP):POKES+5,AD:POKES+6,SR
925 POKES+3,PA:POKES+2,PB:POKES+4,FO
930 FORQ=0TODU:NEXT:POKES+4,FO-1:POKE55960,8:RETURN
```

PROGRAMA COMPROBADO commodore Magazine

```
950 POKE56016,8:POKE56056,8:POKE56096,8:POKE56136,8:RETURN
960 POKECO+XA+(40*YA),8
965 POKECO+XP+(40*YP),3
970 XA=XP:YA=YP:RETURN
975 ES=CD:FORA=0TO3:GOSUB995
980 GETN$:IFN$=""THEN980
985 IFASC(N$)<48ORASC(N$)>57THEN980
990 POKECD,ASC(N$):CD=CD+1:NEXT
992 FORA=0TO3:NU(A)=INT((PEEK(ES+A)-48)*10↑(3-A)):NEXT
994 NT=INT(NU(0)+NU(1)+NU(2)+NU(3)):CD=ES:IFNT>4095THEN975:RETURN
995 FORB=0TO4:POKEES+40+B,32:NEXT:POKEES+40+A,30:RETURN
1000 PRINT"□S=54272                        □FAG□□"
1005 PRINT"VOLUMEN............. POKES+24, 15□□"
1010 PRINT"ALTA FRECUENCIA..... POKE S+1,"INT(AF((XP/2),YP))
1015 PRINT"BAJA FRECUENCIA..... POKE S+0,"INT(BF((XP/2),YP))
1020 PRINT"ATAQUE/DECAIMIENTO.. POKE S+5,";AD
1025 PRINT"SOSTENI/RELAJACION.. POKE S+6,";SR
1030 IFFO<>65THEN1045
1035 PRINT"PULSO ALTO.......... POKE S+3,";PA
1040 PRINT"PULSO BAJO.......... POKE S+2,";PB
1045 PRINT"FORMA DE ONDA....... POKE S+4,";FO
1047 PRINT"□□DURACION NOTA.......";DU
1050 PRINT"□□□□□□PULSE UNA TECLA"
1090 GETA$:IFA$=""THEN1090
1100 RETURN
1200 PRINT"□HASTA LA PROXIMA OCASION                    □FAG□□□□□□□□□□"
1250 PRINT"PULSA RUN STOP/RESTORE□"
1300 POKE650,0:POKE808,235
1450 END
2000 DATA1,18,1,35,1,52,1,75,1,90,1,110,1,132,1,155,1,179,1,205,1,233,2,6,2,37
2005 DATA2,69,2,104,2,140,2,179,2,220,3,8,3,54,3,103,3,155,3,210,4,12,4,73
2010 DATA4,139,4,208,5,25,5,103,5,185,6,16,6,108,6,206,7,53,7,163,8,23,8,147
2015 DATA9,21,9,159,10,60,10,205,11,114,12,32,12,216,13,156,14,107,15,70,16,47
2020 DATA17,37,18,42,19,63,20,100,21,154,22,227,24,63,25,177,27,56,28,214
2025 DATA30,141,32,94,34,75,36,85,38,126,40,200,43,52,45,198,48,127,51,97
2030 DATA54,111,57,172,61,126,64,188,68,149,72,169,76,252,81,161,86,105,91,140
2035 DATA96,254,102,194,108,223,115,88,122,52,129,120,137,43,145,83,153,247
2040 DATA163,31,172,210,183,25,193,252,205,133,217,189,230,176,244,103,0,0
5000 PRINT"□ATAQUE          □F1□":GOSUB5100:PRINT"DECAIMIENTO      □F3□":GOSUB5100
5002 PRINT"SOSTENIMIENTO □F5□":GOSUB5100:PRINT"RELAJACION       □F7□":GOSUB5100
5004 PRINT"□"
5010 PRINT"□"TAB(24)"0 1 2 3 4 5 6 7":PRINT"□"
5020 FORA=0TO11:PRINTTAB(24)"● ● ● ● ● ● ● ●":NEXT
5030 PRINT"□"TAB(21)"□□C□□C#□□D□□D#□□E□□F□□F#□□G□□G#□□A□□A#□□B"
5040 PRINT"□□□FORMA DE ONDA□                  □←□ TOCAR NOTA□"
5042 PRINT"□T□□ TRIANGULO                   □↑□ POKES NOTA"
5045 PRINT"□S□ SIERRA":PRINT"□R□ RUIDO              □=□ TERMINA       "
5047 PRINT"□P□ PULSO":POKENA,134
5050 PRINT"□□A□NCHURA DE PULSO □0000□    □D□URACION □0250":POKENA+1,129
5055 PRINT"□    (0-4095)        ↑       NOTA       ↑□":POKENA+2,135
5100 PRINT"_______________":PRINT"||||||||||||||||":PRINT"_______________"
5105 POKECO,3:RETURN
```

Programas

INVIERTE MEDIA PANTALLA

Con rutinas en lenguaje máquina no pueden hacerse muchas más cosas que con rutinas en BASIC o en cualquier otro lenguaje. Pero hay una diferencia y es que el lenguaje máquina es el campeón de velocidad de los lenguajes (por supuesto velocidad de ejecución, pues a la hora de escribir programas es sin duda alguna el que más tiempo de programación consume).

Esta rutina que os presentamos y que es para el VIC-20 es un ejemplo de lo que acabamos de decir. Se trata de una rutina en lenguaje máquina (aunque se carga en memoria a través de un cargador BASIC).

La rutina permite invertir la mitad superior de la pantalla dejando intacta la mitad inferior de la misma. Concretamente quedan invertidas las 11 primeras líneas.

El cargador almacena la rutina en la zona del buffer del cassette, comenzando en la dirección 832, de forma que una vez cargada, y mientras no se utilice el cassette, la rutina estará a salvo hasta que se apague el ordenador.

Para acceder a la rutina desde un programa BASIC hay que utilizar la siguiente sentencia: SYS (832)

La aplicación más obvia de esta rutina es para programas de juegos, donde todo tipo de efectos extraños son bienvenidos. Sin embargo, seguro que entre nuestros lectores habrá quienes encuentren otro tipo de aplicaciones. ¡Si encontráis alguna realmente interesante hacérnoslo saber!

```
10 N=832
20 READD:IFD=-1THEN80
30 POKEN,D:N=N+1:GOTO20
40 DATA162,242,189,255,29,73
50 DATA128,157,255,29,202,208
60 DATA245,169,6,162,242,157
70 DATA255,149,202,208,250,96,-1
80 PRINT"[illegible]PULSA UNA TECLA"
90 GETT$:IFT$=""THEN90
100 SYS832:GOTO90
```

MAQUINA DE ESCRIBIR

El ordenador tiene un aspecto similar a una máquina de escribir. Así que, ¿por qué no utilizarlo como tal en un momento dado?

El funcionamiento es simple, se carga el corto programa del final, se ejecuta y conectamos la alimentación a la impresora.

El principal inconveniente radica en que no podemos escribir líneas de más de 75 caracteres de longitud. Presionando la tecla de retorno del carro (Return) al final de cada línea hace que ésta sea impresa.

El característico símbolo utilizado por INPUT, la ?, aparece al comienzo de cada línea, pero puede ser evitado con otra breve rutina previamente publicada en **Commodore Magazine.**

Para terminar con el programa simplemente hay que escribir FIN y Return.

```
10 REM**PONER MAYUSCULAS/MINUSCULAS
12 REM**CON TECLAS CBM+SHIFT
20 OPEN3,4,7
30 PRINT#3,CHR$(15)"MAQUINA DE ESCRIBIR";CHR$(13)
40 INPUT X$
50 IF X$="FIN" THEN GOTO 100
60 PRINT#3,X$
70 X$=" "
80 GOTO 40
100 PRINT#3
110 CLOSE3
```

GUARDANDO CEROS Y UNOS

Existe una manera fácil de guardar (SAVE) programas en código máquina en el **Vic-20** ó el **C-64** sin tener que utilizar un largo programa, también en código máquina. Se trata de teclear un par de comandos directos.

Tenemos un programa depositado en una zona de la memoria cuya primera dirección es P y termina en la F. El procedimiento es el siguiente:

PRINT(P/256 - INT(P256))*256, INT(S/256)

Esto produce dos números que llamaremos LP y HP respectivamente.

Tecleemos a continuación:

PRINT(F256 INT(F/256))*256,INT(F/256)

que imprime otros dos números que llamaremos LF y HF.

Ahora entramos en la parte del truco que guarda en almacenamiento magnético el programa:

POKE 43,LP:POKE 44,HP:POKE 45,LF:POKE 46,HF:SAVE «nombre del programa»

Obviamente, las variables LP, HP, LF, HF que aparecen en esta línea son los números que hemos calculado previamente. La línea está pensada para trabajar con *cassette,* pero si utilizamos la unidad de *diskette* la sintaxis es la tradicionalmente utilizada con SAVE para guardar programas.

El comando VERIFY también puede utilizarse para verificar que el programa ha sido almacenado correctamente. Para volver al funcionamiento normal se utilizará SYS 64824 para el **Vic-20** ó SYS 64738 para el **C-64.**

FILTROS PARA ENTRADAS

Cuando desarrollamos programas en los que se deben introducir datos desde el teclado es conveniente establecer una serie de controles. De otra manera es posible que el usuario pueda teclear un tipo de datos no idóneo. El filtro es bastante sencillo y evita que el programa se detenga por estar trabajando con algo que no espera.

Por ejemplo, si el programa necesitase que en un momento dado se presione la tecla A, no continuando a menos que así se haga, completaríamos la línea de la siguiente forma:

100 INPUT A$:IF A$ < > "A" then 100, siendo 100 un número de línea supuesto.

Si, por ejemplo, quisiéramos hacer una introducción de datos numéricos comprendidos entre 15 y 20, la línea se escribiría:

IF A < 15 OR A > 20 THEN 100

Por supuesto que hay variantes, pero ésas serán ideadas y perfeccionadas por el lector.

```
100 INPUT A$:IFA$<>"A"THEN 100
110 PRINT"CORRECTO ":END
```

```
100 INPUT A:IF A<15 OR A>20 THEN 100
110 PRINT"CORRECTO":END
```



SUPERVIVENCIA

Un juego para que vivas la aventura con tu C-64

Volabas con tu avioneta sobre una zona de selva tropical, en un paisaje dominado por el intenso color verde de la vegetación. De repente, un súbito destello de luz, un brillante pero breve destello proveniente de un pequeño claro de la jungla, capta tu atención. Irresistiblemente decides acercarte a investigar y comienzas a volar en círculos sobre la zona hasta que divisas un claro, una explanada de amplitud suficiente como para intentar aterrizar. ¿Cuál será el origen del misterioso destello de luz?.

A medida que te acercas al claro te das cuenta de que la visibilidad empeora, una extraña niebla envuelve tu aparato. Intentas remontarte pero es demasiado tarde, algo ha rozado el ala de la avioneta y...

Cuando comienzas a recuperar el sentido y las ideas van ordenándose en tu cabeza, te das cuenta de que estás encima del panel de mandos, magullado, dolorido, pero milagrosamente entero y con todos los huesos en su sitio.

Después de forcejear durante un rato consigues salir de la cabina y te quedas de pie, en la jungla y contemplando tu maltrecha avioneta. Los daños son de consideración, ni pensar en poder arreglarlos, vas a necesitar ayuda. Lo primero que haces es comprobar si la radio sigue funcionando. Después de unos cuantos intentos consigues enlazar con una base aérea. Realmente parece que estás de suerte.

Explicada tu situación, te contestan que te quedes donde estás, ya que van a enviar una patrulla de ayuda a buscarte. O.K. Todo marcha bien. Quizá

Rutinas de supervivencia

Lo que sigue son unas notas explicando cada una de las rutinas de esta primera parte de SUPERVIVENCIA. Cada una de las rutinas está destinada a una misión concreta. Muchas de ellas han sido nombradas con el verbo que las hace entrar en funcionamiento, por ejemplo «abrir» o «beber», lo cual puede ser de gran ayuda para los que no sepan con qué palabras deben ir avanzando por la jungla.

2-6 ______ *Inicialización de las principales variables del juego. Estas variables van a controlar la situación del jugador a través del juego, el número de objetos que lleva consigo, etc.*

8-28 ______ *Subrutina llamada siempre que el ordenador espera un comando. Es llevada a cabo una limpieza de cadenas de caracteres no utilizadas (garbage colection). Se imprime el mensaje en rojo «AHORA que» y se llama a otra subrutina que permite la entrada del comando en sí. Divide lo introducido en nombre y verbo y pasa a comprobar ambos en el vocabulario del programa.*

30-46 ______ *Se imprime el asterisco de preparado y tiene lugar la entrada y verificación del comando del jugador.*

48-76 ______ *Esta subrutina examina la posición del jugador a través del juego, ajusta los objetos e imprime el mensaje adecuado a la ubicación.*

78-90 ______ *Subrutina llamada desde las rutinas de SAVE o LOAD y que permite saber si está trabajando con cassette o diskette, seleccionando el número de dispositivo adecuado.*

92-98 ______ *Comienzo del juego.*

```
1 REM***SUPERVIVENCIA****
2 CLR
4 CP=2:PD=0:ZZ=1:CH=0:CL=0:DO=0:PA=0:BO=0:GR=0:BT=0:OA=0:HE=0:DI=0:PH=0
6 RN=100:WA=0:BX=0:CN=0:GL=0:MU=0:TM=0:C$=CHR$(13):GOTO92
8 XZ=FRE(0):PRINT:PRINT"[illegible] QUE?[illegible]";:GOSUB30:PRINT"[illegible]"
10 NO$="":VB$="":VB=0:NO=0
12 LC=LEN(CM$):FORI=1TOLC:IFMID$(CM$,I,1)<>" "THENVB$=VB$+MID$(CM$,I,1):NEXT
14 V1$=VB$:VB$=LEFT$(V1$,3):FORI=1TONV:IFVB$(I)=VB$THENVB=I:GOTO18
16 NEXT
18 IFLEN(V1$)+1>=LEN(CM$)THENNO=0:RETURN
20 NO$=RIGHT$(CM$,LEN(CM$)-1-LEN(V1$))
22 N1$=NO$:NO$=LEFT$(NO$,3):FORI=1TONN:IFNO$=NO$(I)THEN28
24 NEXTI
26 NO=0:RETURN
28 NO=I:RETURN
30 CM$=""
32 PRINT"*[illegible]";
34 GETZ$:IFZ$=""THEN34
36 Z=ASC(Z$):IFZ>95THEN34
38 ZL=LEN(CM$):IFZL>27THEN42
40 IFZ>31THENCM$=CM$+Z$:PRINTZ$;:GOTO32
42 IFZ=13ANDZLTHENPRINT" [illegible]":RETURN
44 IFZ=20ANDZLTHENCM$=LEFT$(CM$,ZL-1):PRINTZ$;
46 GOTO32
48 PRINT"[illegible]"
```

—piensas— lo mejor sea descansar hasta que vengan a por mí. Pero después vuelve a tu memoria el misterioso destello de luz y algo dentro de ti te obliga a levantarte. Desde luego —te dices— esto hay que investigarlo...

Así comienza esta aventura en BASIC para el C-64. Si quieres desvelar los misterios que encierra la jungla, lo que tienes que hacer es empezar a teclear el programa. No seas impaciente, ya que hasta el mes que viene no podrás acabar de teclearlo y empezar a jugar. ¿Por qué?, pues porque las dimensiones de este programa nos han decidido a publicarlo en dos partes, la primera de las cuales es ésta que lees, mientras que la segunda podrás encontrarla en el próximo número de *Commodore Magazine.*

El juego en sí es un júego de aventuras en el que tendrás que ingeniártelas para sobrevivir. Para ello tendrás que decirle al ordenador, en todo momento, qué es lo que quieres hacer, y tienes que decírselo escribiendo frases en castellano, del tipo «VE NORTE» o «COGE CUERDA» (algo sencillo, parecido al lenguaje de los indios Sioux).

De este modo irás avanzando por la jungla, descubriendo misterios y evitando peligros, y con un importante objetivo, sobrevivir, evitando un súbito y desagradable encuentro con la muerte.

No te entretenemos más, sabemos que estás ansioso por empezar a teclear, así que adelante. ¡Que tengas suerte! La vas a necesitar.

Rutinas de supervivencia

100 ——— *Si el programa tropieza o se pulsa por error la tecla RUN/STOP esta línea permite recomenzar el juego escribiendo un GOTO.*

102-106 ——— *Llama a la subrutina para la entrada de comandos e imprime mensajes cuando el comando no es comprendido.*

108-112 ——— *Dirige el flujo del programa hacia una subrutina apropiada, según el comando introducido, que se encarga del verbo.*

114-854 ——— *Esta parte del programa cubre el conjunto de rutinas de los verbos. Estas rutinas son examinadas teniendo en cuenta tanto el verbo, como el nombre, la ubicación del jugador, los objetos que lleva consigo y cualquier otro factor relevante. Vamos a examinar cada una de estas rutinas con más detalle.*

114-120 ——— *Rutina de ABANDONAR. Al abandonar el juego se llama a esta rutina que permite guardar el juego en un punto determinado para proseguir, más adelante, a partir de dicho punto.*

122-128 ——— *Rutina de AYUDA. Llamada cada vez que el jugador está perdido y solicita ayuda del ordenador.*

130 ——— *Rutina de REVISAR. Lo que hace es saltar a la rutina que imprime mensajes según la ubicación.*

132-140 ——— *Rutina de INVENTARIO. Si la variable relevante de dentro de la matriz entera tiene el valor de —1, entonces el objeto concreto está en tu posesión. Esta rutina permite conocer qué es lo que llevas encima, en cualquier momento del juego.*

142-144 ——— *Rutina de PUNTOS. Si solamente has llegado a la rutina 21 es que no has estado intentándolo con el suficiente interés.*

146-178 ——— *Rutina de SOLTAR. Evita duplicaciones (que cojas dos veces el mismo objeto, por ejemplo), comprueba que tienes el objeto y llama a la acción apropiada para dejar caer determinados objetos en*

```
50 IFCP<24ORCP>56THENPD=0:GOTO56
52 IF(CP>49ANDCP<56)ANDOB%(30)<>-1THENPRINT"♥IN UN BOTE TE HAS AHOGADO.":GOTO130
6
54 IFOB%(23)<>-1THENPD=1:PRINTPD$:POKE53280,11:POKE53281,12:RETURN
56 PRINT"¯STAS ";P$(CP)
57 IFCP=84THENGOTO1312
58 IFPH=0ANDCP=83THENPRINTPH$
60 IFCP=13ANDOB%(18)<>-1THENPRINTUP$:GOSUB1290:GOTO98
62 IFCP=62THENGOTO1306
64 :
65 IFCP=81ORCP=82THENGOSUB1290:CP=83:GOTO98
66 VB$="▀▜UEDES VER :-                    "
68 IFP%(29,2)ANDCP=29THENOB%(29)=29
70 IFP%(30,3)ANDCP=30THENOB%(29)=30
72 FORI=1TOLO:IFOB%(I)=CPANDOB$(I)<>""THENPRINTVB$;OB$(I):VB$=""
74 NEXTI
76 RETURN
78 PRINT"▀▀¯STAS UTILIZANDO CINTA ? (♥/╱)"
80 GETA$:IFA$=""THEN80
82 IFA$="S"THEN86
84 DEV=8:RETURN
86 PRINT"▜╲NTRODUCE LA CINTA Y PULSA ▛¯♥▜♠¬┌▀."
88 GETA$:IFA$<>" "THEN88
90 DEV=1:RETURN
92 PRINT"▜";CHR$(14):POKE53280,5:POKE53281,7
94 PRINT"▀▀▀▀▀▀▀▀¯SPERA UN POCO A QUE ME ORIENTE"
96 GOSUB876
98 GOSUB48
100 IF(CP<24ORCP>56)OR(OB%(23)=-1OROB%(23)=CP)THENPOKE53280,5:POKE53281,7
102 GOSUB8
104 IFVB>5ANDNO$=""THENPRINTV1$;" ♥/¯!?":GOTO100
106 IFV1$<>""ANDVB=0THENPRINT"╱O ENTIENDO,"V1$:GOTO100
108 ONVBGOTO114,122,130,132,142,146,180,232,260,264,268,316,334,350,360,372
110 ONVB-16GOTO380,384,406,422,478,528,540,550,566,628,636,646,666
112 ONVB-29GOTO678,690,726,738,756,766,790,810,830,846
114 PRINT"┌╱.":PRINT"▀▀UARDAS EL JUEGO EN ESTE PUNTO? (♥/╱)"
116 GETA$:IFA$="S"THEN766
118 IFA$<>"N"THEN116
120 END
122 IFCP=13THENPRINTBA$:GOTO100
124 IFCP=70THENPRINTGN$:GOTO100
126 PRINT"!ENGO MEJORES COSAS QUE HACER QUE AYUDARTE CON EL JUEGO!"
128 PRINT">AMOS HOMBRE,SIGUE!":GOTO100
```

determinadas situaciones. Se decrementa la variable de número de objetos zz.

180-230 —— *Rutina de IR. Comprueba si eres capaz de irte examinando el estado de determinados indicadores (flags). ¿Dispones de los objetos adecuados?, ¿llevas demasiados objetos?, etc.*

260-263 —— *Rutina de CRUZAR. Ayuda al jugador sobre un obstáculo donde un comando de dirección no lo haría.*

264-266 —— *Rutina de TOMAR. Asume que estás de juerga e ignora el comando, soltándote una frasecita ingeniosa.*

268-314 —— *Rutina de ABRIR. Si se cumplen todas las condiciones (para ello se examinan una serie de indicadores) la cosa se abre, se quita el bloqueo de la cadena y se cambia la descripción de cualquier ubicación u objeto.*

316-322 —— *Rutina de CERRAR.*

334-348 —— *Rutina de COMER. Llamada cada vez que quieras comer algo.*

350-358 —— *Rutina de ALIMENTAR. Es un poco diferente a la de comer, ya que se trata de dar de comer a alguien o a algo distintos del jugador.*

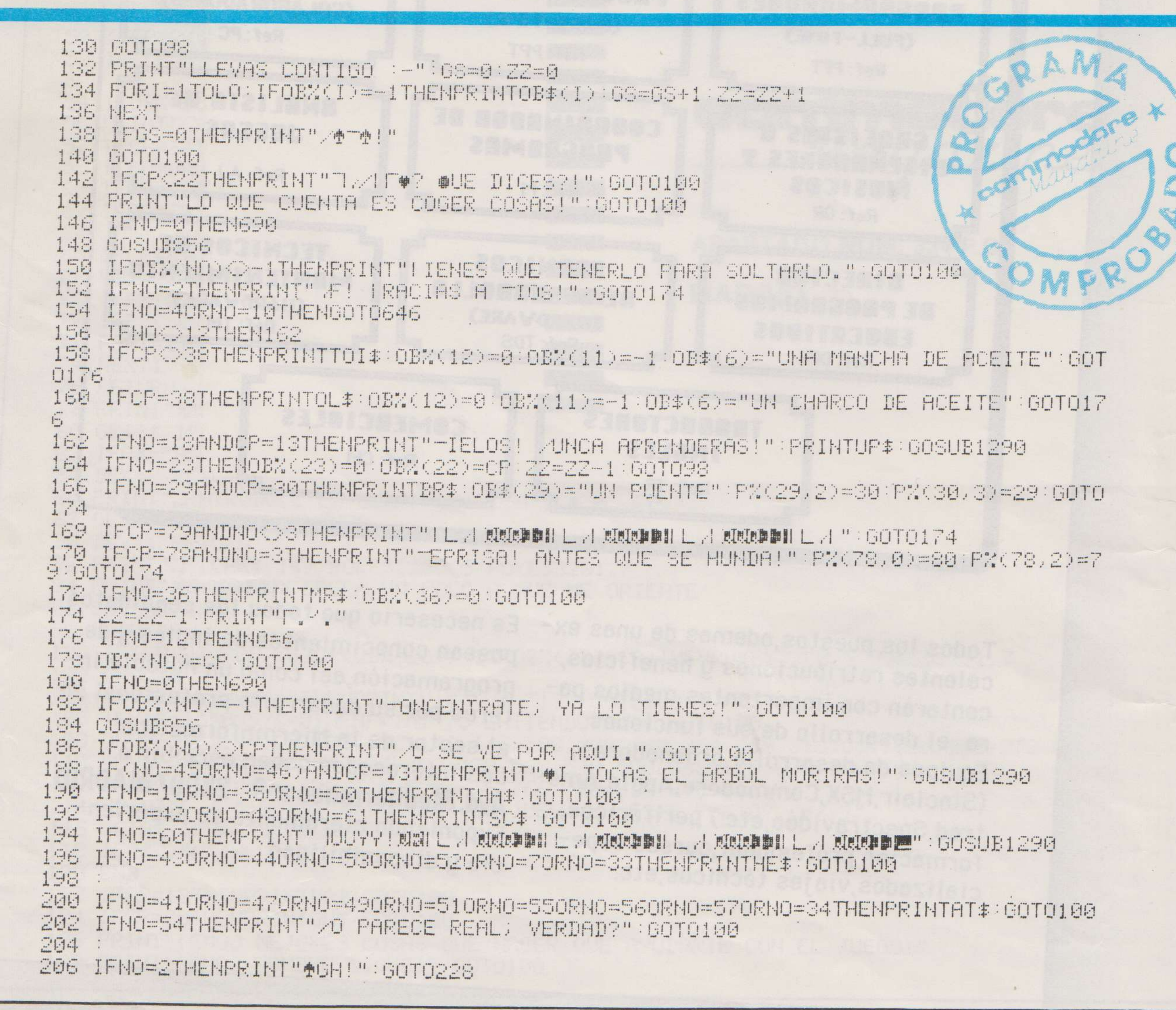
360-370 —— *Rutina de BEBER.*

372-378 —— *Rutina de OFRECER. El comando es tomado a broma excepto en una ocasión.*

380-382 —— *Rutina de BALANCEAR. No es bueno hacerlo en este juego, pero siempre se puede intentar, a ver qué pasa.*

384-404 —— *Rutina de CORTAR.*

406-420 —— *Rutina de TROCEAR. Existe una ligera diferencia con la acción de cortar. Esta asume que eres un poco manazas, mientras que cortar implica un cierto grado de entrenamiento.*



```
130 GOTO98
132 PRINT"LLEVAS CONTIGO :-":GS=0:ZZ=0
134 FORI=1TOLO:IFOB%(I)=-1THENPRINTOB$(I):GS=GS+1:ZZ=ZZ+1
136 NEXT
138 IFGS=0THENPRINT"/♠¯♠!"
140 GOTO100
142 IFCP<22THENPRINT"⌝/|Γ♥? ♥UE DICES?!":GOTO100
144 PRINT"LO QUE CUENTA ES COGER COSAS!":GOTO100
146 IFNO=0THEN690
148 GOSUB856
150 IFOB%(NO)<>-1THENPRINT"|IENES QUE TENERLO PARA SOLTARLO.":GOTO100
152 IFNO=2THENPRINT" ╱F! |RACIAS A ¯IOS!":GOTO174
154 IFNO=4ORNO=10THENGOTO646
156 IFNO<>12THEN162
158 IFCP<>38THENPRINTTOI$:OB%(12)=0:OB%(11)=-1:OB$(6)="UNA MANCHA DE ACEITE":GOT
0176
160 IFCP=38THENPRINTOL$:OB%(12)=0:OB%(11)=-1:OB$(6)="UN CHARCO DE ACEITE":GOTO17
6
162 IFNO=18ANDCP=13THENPRINT"-IELOS! /UNCA APRENDERAS!":PRINTUP$:GOSUB1290
164 IFNO=23THENOB%(23)=0:OB%(22)=CP:ZZ=ZZ-1:GOTO98
166 IFNO=29ANDCP=30THENPRINTBR$:OB$(29)="UN PUENTE":P%(29,2)=30:P%(30,3)=29:GOTO
174
169 IFCP=79ANDNO<>3THENPRINT"|L/▒▒▒▒|L/▒▒▒▒|L/▒▒▒▒|L/":GOTO174
170 IFCP=78ANDNO=3THENPRINT"¯EPRISA! ANTES QUE SE HUNDA!":P%(78,0)=80:P%(78,2)=7
9:GOTO174
172 IFNO=36THENPRINTMR$:OB%(36)=0:GOTO100
174 ZZ=ZZ-1:PRINT"Γ.'."
176 IFNO=12THENNO=6
178 OB%(NO)=CP:GOTO100
180 IFNO=0THEN690
182 IFOB%(NO)=-1THENPRINT"¯ONCENTRATE, YA LO TIENES!":GOTO100
184 GOSUB856
186 IFOB%(NO)<>CPTHENPRINT"/O SE VE POR AQUI.":GOTO100
188 IF(NO=45ORNO=46)ANDCP=13THENPRINT"♥I TOCAS EL ARBOL MORIRAS!":GOSUB1290
190 IFNO=1ORNO=35ORNO=50THENPRINTHA$:GOTO100
192 IFNO=42ORNO=48ORNO=61THENPRINTSC$:GOTO100
194 IFNO=60THENPRINT" ¦UUYY!▒▒|L/▒▒▒▒|L/▒▒▒▒|L/▒▒▒▒|L/▒▒▒▒":GOSUB1290
196 IFNO=43ORNO=44ORNO=53ORNO=52ORNO=7ORNO=33THENPRINTHE$:GOTO100
198 :
200 IFNO=41ORNO=47ORNO=49ORNO=51ORNO=55ORNO=56ORNO=57ORNO=34THENPRINTAT$:GOTO100
202 IFNO=54THENPRINT"/O PARECE REAL, VERDAD?":GOTO100
204 :
206 IFNO=2THENPRINT"♠GH!":GOTO228
```

Frases del listado

Supervivencia trabaja fundamentalmente con frases. Estas frases y palabras van a aparecer escritas tanto en mayúsculas como en minúsculas. Para ello el programa se pone en modo minúsculas y en las sentencias PRINT, las letras mayúsculas se teclean pulsando al mismo tiempo la letra y la tecla de SHIFT. Es por esta razón por lo que en el listado aparecen caracteres extraños dentro de las sentencias PRINT, y por lo que acompañamos una lista de las instrucciones en las que aparecen estos caracteres gráficos y la palabra o frase con que se obtienen. Por ejemplo la primera línea es la 8, en la que aparecen unos caracteres gráficos y luego la palabra «que» dentro de una sentencia PRINT. Pues bien, consultando la lista se puede ver que los caracteres gráficos corresponden a la palabra AHORA, en mayúsculas, esto es pulsando todas las teclas de la palabra junto con la tecla SHIFT.

8	AHORA qué?
52	Sin un bote...
56	Estas
66	Puedes ver...
78	Estás utilizando cinta? (S/N)
86	Introduce... y pulsa ESPACIO
94	Espera...
104	QUE?
106	No entiendo
114	Guardas el juego en este punto? (S/N)
126	Tengo...
128	Vamos...

```
208 IFNO=4ANDOB%(21)<>-1THENPRINT"-OMO? EN LA MANOS?":GOTO100
209 IFNO=4ANDOB%(21)=-1THENOB$(21)="UNA PALETADA DE BARRO":OB%(4)=0:PRINT"Γ./."G
OTO100
210 IFNO=6ANDOB%(11)<>-1THENPRINT"7N QUE?":GOTO100
212 IFNO=6ANDOB%(11)=-1THENOB%(11)=0:OB%(6)=0:NO=12:ZZ=ZZ-1:GOTO228
214 IFNO=17THENPRINTCO$:OB$(17)="UN GIRO COMPAS":GOTO100
216 IFNO=21ANDOB%(32)<>-1THENPRINTFA$:GOTO100
218 IFNO=26THENPRINTDU$:OB%(26)=0:OB%(27)=CP:GOTO100
220 IFNO=29AND(CP=29ORCP=30)THENPRINTJA$:GOTO100
222 IFNO=37THENGOSUB1286:GOTO100
224 IFNO=39THENPRINTBU$:GOTO100
226 IFNO=38THENPRINT"/O, NO, /Γ!":GOTO100
228 ZZ=ZZ+1:IFZZ>5THENPRINT"/O PUEDES LLEVAR NADA MAS!":ZZ=ZZ-1:GOTO100
230 OB%(NO)=-1:PRINT"|A ESTA!":GOTO100
232 DI=0
234 IFNO<62ORNO>69THENPRINT"♠ DONDE?":GOTO100
236 IFNO>65THENNO=NO-4
238 NO=NO-62
240 IFOB%(16)<>-1THENPRINTDI$:GOTO100
242 IFNOANDPDTHENPRINT"|E LO DIJE, HAS CAIDO EN UN POZO!":GOTO1290
244 IFP%(CP,NO)=0THENPRINT"/O PUEDES IR EN ESA DIRECCION.":DI=1
246 IF(CP=2ANDNO=1)ANDHE<>1THENPRINT"|ODAVIA!":GOTO100
248 IF(CP=13ANDNO=2)ANDHO<>2THENPRINT"|ODAVIA!":GOTO100
250 IF(CP=18ANDNO=3)ANDMU<>1THENPRINT"|ODAVIA!":GOTO100
252 IF(CP=56ANDNO=0)ANDOB%(30)=-1THENPRINT"7L BOTE NO PASA DE AQUI.":GOTO100
254 IF(CP=78ANDOB%(3)<>CP)AND(NO=0ORNO=2)THENPRINT"|LUG▒▒▒|LUG▒▒▒|LUG▒▒▒|LUG.":G
OTO1306
256 IFDI=0THENCP=P%(CP,NO):GOTO98
258 GOTO100
260 IFNO=0THEN690
262 PRINT"♠ NO SER QUE PROYECTES ALGO MUY ORIGINAL, NO FUNCIONARA!"
263 PRINT"7IENSA OTRA COSA!":GOTO100
264 IFNO=0THEN690
266 PRINT"♠CASO ESTAS ENFERMO?":GOTO100
268 IFNO=0THEN690
270 GOSUB856
272 IFNO=47THENPRINT"7S INUTIL!":GOTO100
274 IFNO=5THENPRINT"/O. ♠SI NO LO CONSEGUIRAS!":GOTO100
276 IFNO=56THEN302
278 IFNO=7THEN294
280 IFNO=57THEN550
282 IFNO=33ORNO=34THEN312
284 IFNO<>38THENPRINT"/O HAY NADA QUE ABRIR!":GOTO100
286 IFDO=1THENPRINT"|A ESTA ABIERTA!":GOTO100
288 IFOB%(28)<>-1THENPRINT"-ON QUE? -ON UNA HORQUILLA?":GOTO100
290 PRINT"LA PUERTA SE ABRE DEJANDO VER UNA HABITACION.":P%(47,1)=49
292 OB$(38)="UNA PUERTA ABIERTA":DO=1:GOTO100
294 :
296 :
298 :
300 GOTO100
302 IFOB%(28)<>-1THENPRINT"/O TIENES CON QUE ABRIR.":GOTO100
```

SU PROGRAMA PARA CUALQUIER SISTEMA COMMODORE PUEDE HACERLE GANAR 5.000 PTAS.

EL PRESENTE CONCURSO ESTA ABIERTO A TODOS NUESTROS LECTORES Y SU PARTICIPACION E INSCRIPCION ES GRATUITA.
LEA LAS BASES DEL CONCURSO

- NO SE ESTABLECEN LIMITACIONES EN CUANTO A EXTENSION, TEMA ELEGIDO O MODELO DE ORDENADOR
- LOS CONCURSANTES DEBERAN ENVIARNOS A LA DIRECCION QUE FIGURA AL PIE, EL CASSETTE O DISKETTE CONTENIENDO EL PROGRAMA, UNA EXPLICACION DEL MISMO Y, AL SER POSIBLE, UN LISTADO EN PAPEL DE IMPRESORA. SE PODRAN ENVIAR TANTOS PROGRAMAS COMO SE DESEE
- LOS PROGRAMAS, PREVIA SELECCION, SERAN PUBLICADOS EN LA REVISTA, OBTENIENDO TODOS ELLOS 5.000 PTAS.
- LA DECISION SOBRE LA PUBLICACION O NO DE UN PROGRAMA CORRESPONDE UNICAMENTE AL JURADO NOMBRADO AL EFECTO POR "COMMODORE MAGAZINE", SIENDO SU FALLO INAPELABLE
- LOS CRITERIOS DE SELECCION SE BASARAN EN LA CREATIVIDAD DEL TEMA ELEGIDO Y LA ORIGINALIDAD Y/O SENCILLEZ EN EL METODO DE PROGRAMACION GLOBAL
- ENVIAR A: CONCURSO COMMODORE MAGAZINE

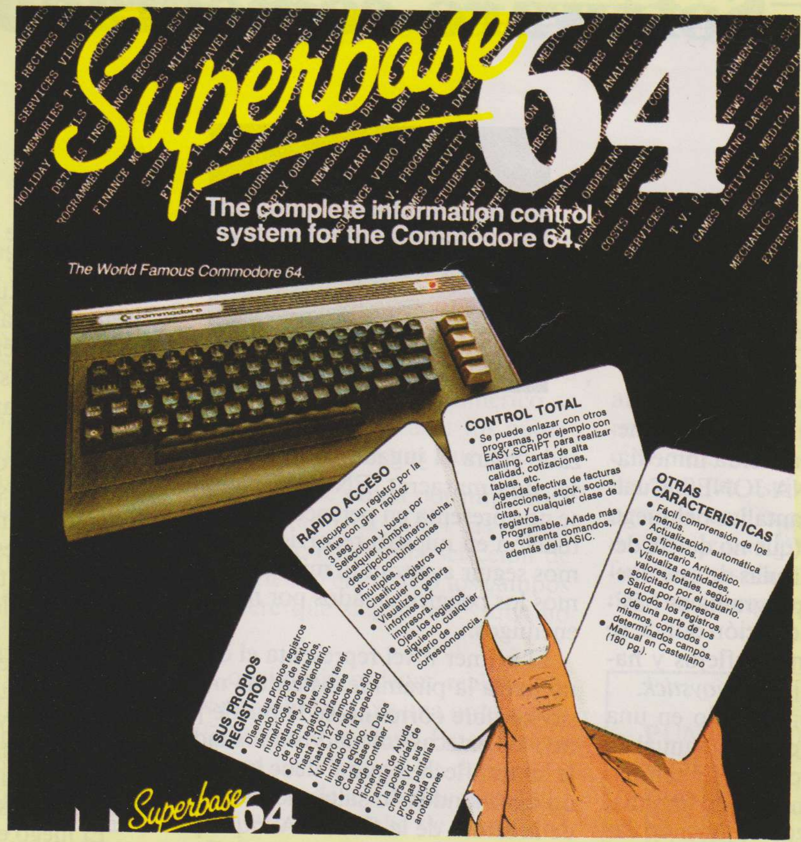


commodore Magazine

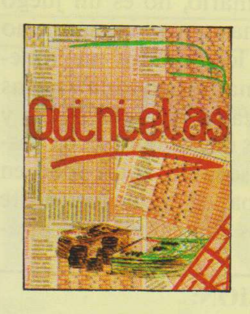
C/Bravo Murillo, 377. 28002-Madrid

132	Llevas...	194	Huuyy! GLUC GLUC GLUC	266	Acaso...
138	NADA	208	Como...	272	Es...
142	PUNTOS? Qué dices?	210	En...	274	No. Así...
144	Lo...	226	No, no, NO!	288	Con qué? Con...
150	Tienes...	228	No...	302	No...
152	Uf! Gracias a Dios!	230	Ya...	324	Ya...
162	Cielos! Nunca...	234	A donde	342	Beeghhh...
169	GLUC GLUC GLUC GLUC	242	Te...	344	Ummm...
170	Deprisa...	246	Todavía	366	Beug...
174	O.K.	252	El...	368	GLUG...Ahhhh.
182	Concéntrate...	254	Glug...	370	A...
186	No...	262	A no ser...	372	Ofrecer QUE?
188	Si...	263	Piensa...	396	Es...

```
304 IFOB%(28)=-1THENPRINT"LOS MUELLES DEL CANDADO CEDEN.":PA=1
306 OB$(7)="UNA VERJA CON CANDADO"
308 IFBO=1THENPRINTGA$:GR=1:P%(50,2)=51:OB$(7)="UNA VERJA ABIERTA"
310 GOTO100
312 :
314 :
316 IFNO=0THEN690
318 GOSUB856
320 IFNO<>7ORNO<>38THENPRINT"/O HAY NADA QUE CERRAR!":GOTO100
322 IFNO=7ANDGR=1THENPRINT"LA FUERZA DEL AGUA IMPIDE QUE SE CIERRE.":GOTO100
324 IFNO=7ANDGR<>1THENPRINT"LA FUERZA DEL AGUA IMPIDE QUE SE CIERRE.":GOTO100
326 IFNO=38ANDDO<>1THENPRINT"!A ESTA CERRADA!":GOTO100
328 IFNO=38ANDDO=1THENPRINT"LA PUERTA SE CERRADA, VALE?!":GOTO100
330 IFDO=0THENOB$(38)="UNA PUERTA CERRADA":DO=0:P%(47,1)=0:OB%(38)=CP
332 GOTO100
334 IFNO=0THEN690
336 IFOB%(NO)<>-1ANDNO<>33THENPRINT"/O TIENES DE ESO.":GOTO100
338 IFNO=37THENGOSUB1286:GOTO100
340 IFNO=2THENPRINT"/O RECUERDAS COMO MURIO?":ZZ=ZZ-1:OB%(2)=0:GOSUB1290:GOTO100
342 IFNO<>33THENPRINT"!EEGHHH. /O ES BUENA IDEA.":GOTO100
344 PRINT"/MMMM...DELICIOSO."
346 PRINT"LA ROCA DESAPARECE EN UNA NEBLINA MOSTRANDO UN CAMINO HACIA DELANTE."
348 OB%(33)=0:P%(28,1)=29:GOTO100
350 IFNO=0THEN690
352 IFNO=37THENGOSUB1286
354 IFOB%(2)<>-1THENPRINT"/O TIENES NADA QUE LE GUSTE.":GOTO100
356 IFNO=52THENPRINTAC$:P%(70,0)=77
358 ZZ=ZZ-1:OB%(2)=0:GOTO100
360 IFNO=0THENPRINT"BEBER QUE?":GOTO100
362 GOSUB856
364 IFNO=6THENPRINT"EH?"GOTO100
366 IFNO=12THENPRINT"!EUG....":OB%(12)=0:OB%(11)=-1:GOTO100
368 IFNO=10THENPRINT"!L/▮▮▮▮▮L/▮▮▮▮▮L/. ♠HHHHH.":OB%(10)=0:OB%(11)=-1:GOTO1
00
370 PRINT"♠ VECES RESÙLTAS TAN ...!!":GOTO100
372 IFNO=0THENPRINT"┌FRECER ●/‾!?":GOTO100
374 IFCP=13THENGOSUB1286:GOTO100
376 IFNO=2ANDCP=70THENNO=52:GOTO350
378 PRINT"/O QUIERE ESO!":GOTO100
380 IFNO=0THEN690
382 PRINT"‾OR ‾IOS! ●UE HACES!":GOTO100
384 IFNO=0THEN690
386 GOSUB856
388 IFOB%(19)<>-1ANDOB%(24)<>-1THENPRINT"-ON QUE DIABLOS?":GOTO100
390 IFCP<>13ANDCP<>46THENPRINT"/O HAY NADA INTERESANTE QUE CORTAR.":GOTO100
392 IFNO=37THENGOSUB1286:GOTO100
394 :
396 IFNO<>42THENPRINT"‾S INUTIL!":GOTO100
398 :
400 :
402 :
404 P%(13,2)=15:OB%(19)=13:ZZ=ZZ-1:OB$(19)="ASTILLAS DE ACERO":GOTO100
```



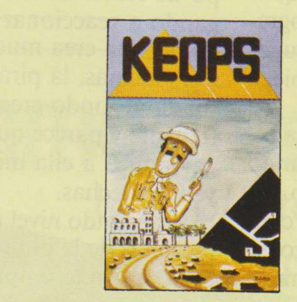
P.V.P. 22.500 Ptas. (disco)



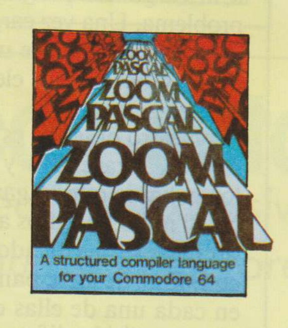
Elimina variantes y figuras. Desarrollo, lista por impresora los boletos del patronato o en papel continuo para archivo, introduciendo el boleto premiado hace el escrutinio de la jornada imprimiendo los premios dándole el número de boleto y columna, etc.
P.V.P. 15.000 Pts.



Usa el mismo intérprete que el Basic del ordenador. Mantiene la misma memoria libre y velocidad de ejecución.
Comandos y mensajes en castellano.
P.V.P. 2.000 Ptas. cinta
P.V.P. 2.500 Pts. disco



Maravilloso juego de aventuras. Deberá emplear su inteligencia y su ingenio.
P.V.P. 1.500 Ptas. cinta
P.V.P. 2.000 Ptas. disco



Zoom-Pascal para CBM-64 extremadamente rápido. Añade de extensiones y funciones que lo hacen verdaderamente útil.
P.V.P. 12.000 Pts.
P.V.P. 12.500 Pts.

CONTROL DE ALMACEN

Controla 1.200 referencias de almacén, en unidades, kilos, litros, metros, etc. Stock mínimo. Realiza gestiones de entrada y salida (incluidas devoluciones). Podrá hacer altas, bajas, modificaciones y consultas, Proceso de Inventario Físico, Inventario a Precio de Costo y a Precio de Venta. Etiquetas...
Le dará en cualquier momento un listado de productos sin movimiento. Además tiene una opción de anotación para los gastos de su empresa, con análisis de rentabilidad de su negocio, sin pasar por contabilidad. De fácil manejo. TODO EN CASTELLANO.
P.V.P. en disco 35.000 Pts.

SAKATI, S.A. Empresa Líder en Soft para COMMODORE-64, tiene un amplio catálogo de programas, donde Vd. encontrará la solución a sus necesidades.

¡CONSULTENOS!

BOLETIN DE PEDIDO

A enviar a SAKATI, S.A. - Ardemans, 24 - 28028-MADRID

Ref.	Cdad.	Precio

☐ Talón adjunto ☐ Contra reembolso TOTAL ________

Fecha: Firma:

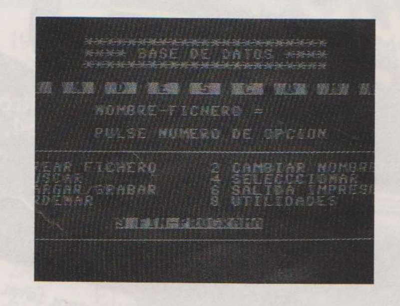
Don ________

Calle ________ N.º ________

Código Postal ________ Ciudad ________

Provincia ________ Tel. ________

Software comentado



PROGRAMA:
AZTEC CHALLENGE
TIPO: JUEGO
DISTRIBUIDOR: ABC SOFT
ORDENADOR: COMMODORE 64
CON JOYSTICK

Cuando se ve este juego por primera vez el jugador se acuerda inmediatamente de INDIANA JONES. Cualquiera de las siete pantallas del juego tiene su equivalente (que no quiere decir plagio) en las películas de este nuevo Super héroe. El programa es básicamente un juego de acción donde se necesitan muy buenos reflejos y habilidad en el manejo del *joystick.*

El juego viene presentado en una atractiva caja junto con un manual de instrucciones muy completo comparado con lo que viene siendo habitual en los últimos juegos. Siguiendo las instrucciones del manual se procede a la carga del programa sin ningún problema. Una vez cargado, el programa deja elegir entre uno y dos jugadores y hecha esta elección empieza el juego.

Nuestro objetivo es entrar en la pirámide, atravesarla y salir, evitando los peligros que salgan a nuestro paso. Si no lo hacemos así seremos capturados y sacrificados. El juego se compone de siete pantallas distintas, en cada una de ellas el *joystick* cumple una misión diferente a la anterior, lo que unido a la dificultad que tiene el juego en sí, lo convierte en un auténtico reto para el jugador avanzado (para el jugador novato, puede ser una masacre). EN cada pantalla se nos presenta en perspectiva una situación en nuestro recorrido y debemos seguir avanzando mientras evitamos los peligros creados por nuestros enemigos.

El primer nivel representa el camino hacia la pirámide. Mientras nuestro hombre corre (no se le puede parar) los aztecas colocados a los lados le lanzan flechas que él debe ir evitando agachándose o saltando, depende de la altura de la flecha. Aunque parece extremadamente sencillo el tiempo de reacción es muy corto y obligando a reaccionar demasiado rápido, lo que crea muchos errores fatales. Además, la pirámide que está situada al fondo crece demasiado lentamente y parece que no vamos a llegar nunca a ella mientras siguen cayendo flechas.

El segundo nivel nos sitúa ya en las escaleras de la pirámide. Mientras vamos subiendo empiezan a caer grandes rocas capaces de aplastarnos. Moviéndonos rápidamente podemos esquivarlas aunque cada vez aumenta el número de piedras que caen y, además, como todo el mundo sabe, las escaleras de las pirámides son extremadamente largas.

Las afortunadas personas que logran subir las escaleras y entran en el templo, se encuentran con múltiples habitaciones protegidas por diversos sistemas. Lanzas que caen del techo, agujeros en el suelo, etc.

Después del templo, y cuando ya parece que se ha terminado el sufrimiento, vienen otros cuatro niveles más difíciles si cabe. En la cámara central del templo nos encontramos con múltiples animales que con su simple roce nos matan. Algunos saltan, otros se arrastran, etc. Debemos saltar y agacharnos para evitarlos y poder salir y atravesar los tres niveles que quedan. En éstos nos encontraremos con un lago con pirañas, puentes mortales, etc.

El juego es un auténtico desafío difícil de ganar que necesita tiempo para poder dominarlo, no es un juego para las personas que se inician, sino más bien para los expertos que quieren probar el más difícil todavía. Las gráficas son realmente fascinantes y el efecto de tres dimensiones está bastante bien conseguido, sobre todo en la primera etapa, en la que realmente parece que nos acercamos a la pirámide.

PUNTUACION:
ADICION: 8
PRESENTACION: 8
GRAFICOS: 8
ACCION: 9

PROGRAMA:
BASE DE DATOS-FICHEROS
TIPO: UTILIDAD
DISTRIBUIDOR: INDESCOMP
FORMATO: CINTA
DE CASSETTE
ORDENADOR: COMMODORE 64

Este programa permite al usuario la creación de sus propios ficheros y el manejo de éstos introduciendo y modificando los datos de los mismos. También permite el listado de todos los datos.

Una vez cargado el programa se muestra un menú de 9 opciones. La primera permite la definición de un fichero. Cada vez que queramos crear un nuevo fichero de datos (por ejemplo, los teléfonos de todos nuestros amigos) hay que decirle al ordenador previamente qué datos hay y de qué tipo. Cada dato puede ser de dos tipos: numérico o caracteres. Los datos numéricos (teléfonos, ingresos, etc.) pueden tener un máximo de 9 números de largo. Los datos del tipo carácter son aquellos que admiten cualquier símbolo que se teclee y no sólo los números. La longitud máxima de estos campos es de 36. Por cada dato distinto que creemos hay que darle tres datos: primero se indica el nombre del dato seguido por el tipo y por último la longitud. Esta longitud puede ser la máxima pero, naturalmente, puede ser menor. Si nos definimos un campo que sea el código postal le llamaremos «CODIGO» su tipo será «N» (numérico) y su longitud será 5, ya que con cinco cifras basta para indicar un código postal. El número

máximo de datos distintos que admite cada ficha es de 15, si intentamos meter más de estos nos encontraremos con que el ordenador nos lo rechaza y vuelve a pedir el número de campos.

La segunda opción nos permite cambiar el nombre del fichero (un fichero que se llama «amigos» podemos llamarlo, si se da la situación «enemigos»).

La opción 3, denominada genéricamente «buscar» permite la actualización de datos del fichero dando altas, bajas, modificando fichas, etc.

Otra opción permite la lectura y grabación de datos en el fichero. Hemos de recordar que si apagamos el ordenador los datos que haya en memoria se borran, con lo que nuestro fichero se va a las nubes si previamente no hemos grabado los datos en la cinta.

Otra opción nos permite buscar una ficha específica, indicando el campo por el que queremos buscar y el dato que buscamos (por ejemplo, en el campo «CODIGO» el número 28015). Si el campo es numérico se nos da la opción de sacar los números mayores y menores que éste.

La opción de salida por impresora saca un listado de todas las fichas (no se puede hacer una selección de ellas) eligiendo el usuario los campos que quiere que salgan y los que no quiere.

La opción de ordenar realiza, como su nombre indica, una ordenación del fichero por la clave. Esta clave es el primer dato que se ha definido y sale separado de los demás cuando se nos muestra una ficha de modo que sea fácilmente visible.

Aparte de las opciones anteriores existe otra que nos indica la memoria libre y una última que hace que el programa termine.

El programa cumple su función aunque se hacen notar algunas opciones (listado selectivo de las fichas, por ejemplo). Las instrucciones son claras aunque están escritas para una persona con experiencia y al nuevo usuario le puede crear una gran confusión que se hubiera podido solucionar sencillamente.

PUNTUACION:
UTILIDAD: 7
PRESENTACION: 5
CLARIDAD: 8
RAPIDEZ: 4



Envianos la foto de tu ordenador

Continúan llegando a nuestra redacción numerosas muestras del ingenio de los commodoreros para ubicar sus ordenadores. Cuatro lectores nos presentan esta vez sus soluciones con unas fotografías de excelente calidad. ¿Quién tiene mejores ideas? Esperamos vuestras cartas.



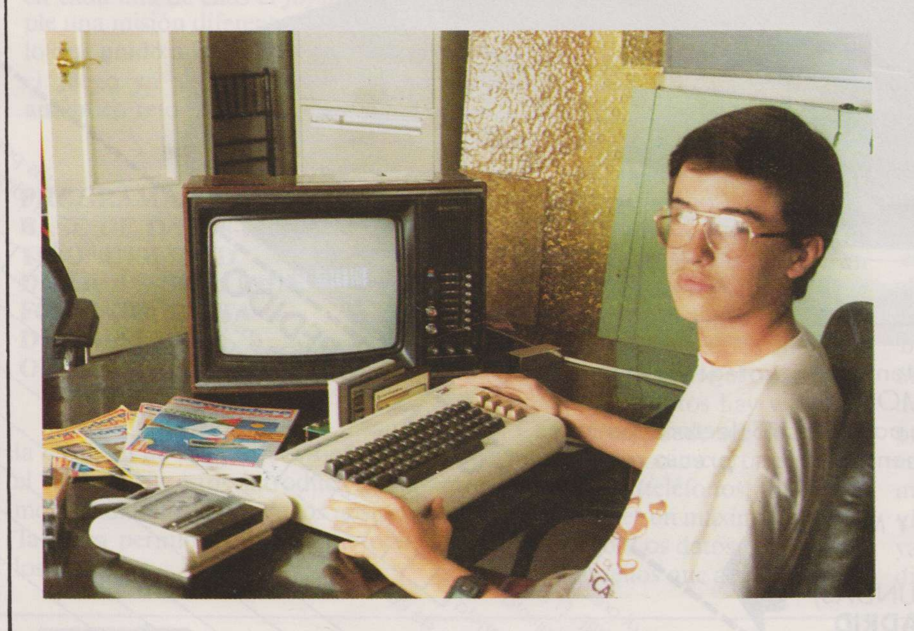




MADRID

Juan José Hernández, de Madrid, ha encontrado una interesante solución en la que se dan la mano lo moderno y lo clásico. Un recio mueble castellano que descubrió en un mercado de segunda mano le sirve de soporte a su instalación, con la peculiaridad de que por tratarse de un escritorio, la cortinilla de madera aísla y protege el teclado cuando no está en uso. En la bandeja que se desliza hacia el exterior está el VIC, quedando a la derecha un espacio para escribir. En uno de los cajones laterales se oculta el cassette. Los cables salen del mueble por un orificio practicado en la tapa posterior. Una vez cerrado sólo queda el monitor en el exterior.

Enhorabuena por esta solución llena de buen gusto que dignifica, dándole un toque rabiosamente actual, un mueble clásico.



SABADELL

Para Pedro, David y Esther Gómez, el ordenador es un tema absolutamente familiar. El gusanillo commodorero les ha atacado en bloque sin distinguir entre mayores y pequeños. En la fotografía que reproducimos a la izquierda y que nos ha sido remitida por Pedro Gómez, podemos ver a los más jóvenes de la casa en el rincón del estudio que su progenitor ha habilitado para las numerosas horas del estudio y práctica que todos dedican a su máxima afición. Padre e hijos rivalizan en ingenio y habilidad al mismo tiempo que se complementan para mejorar los programas. Cuentan con un VIC-20 y un CBM-64 colocados sobre una sencilla mesa, mientras en una repisa han situado la pantalla.

Estupenda manera de «pensar unidos» que sin duda tendrá beneficiosos resultados a la hora de solventar futuros conflictos generacionales.



TARRAGONA

Angel López Romaní, de Tarragona, es un hombre práctico que ha sabido ganarle la partida al sempiterno problema de espacio mediante un inteligente sistema móvil.

A partir de una vieja mesa de televisor y tablas, ha armado un aparato en el que el transformador y todo el cableado están ubicados en un compartimento cerrado, tras el ordenador y bajo el monitor. El compartimento lleva rejillas para refrigeración, desmontadas del ventilador de un frigorífico inservible. Cartuchos y ampliaciones también van en el compartimento de ventilación. Del «mueble» sale un único cable para la toma de corriente.



ALICANTE

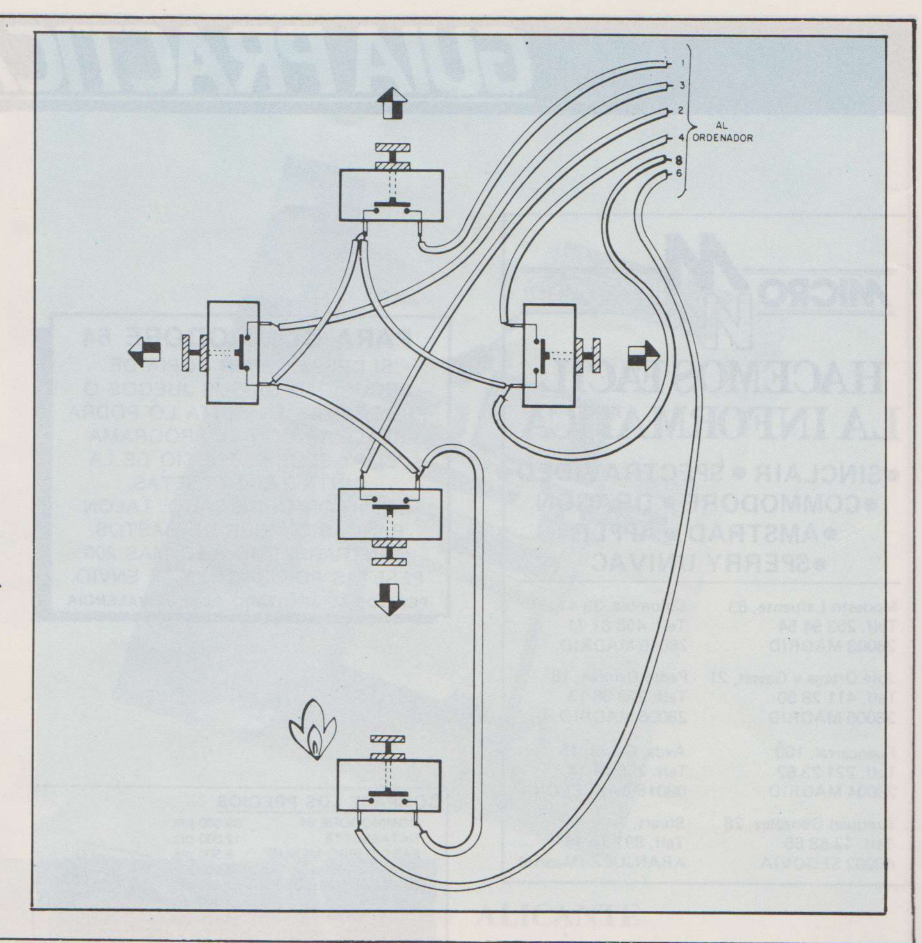
El caso de Vicente Flores Arjente, alicantino de Altea, es la excepción que confirma la regla pues parece disponer de todos los centímetros cuadrados necesarios para desplegar su Commodore VIC-20. Le deseamos que pronto su creciente afición le haga insuficiente el espacio disponible y que los periféricos y accesorios se le amontonen por todas partes. Por lo pronto vemos que va bien encaminado puesto que parece ser un asiduo lector de nuestro Commodore Magazine, si juzgamos por el impresionante despliegue que se puede ver en la fotografía.

GUIA PRACTICA

Aclaración

Construye tu JOYSTICK

En el artículo «Construye tu *joystick*», publicado en el número 8 de **Commodore Magazine** se omitió el hilo común en el dibujo correspondiente a la realización práctica. De todas formas nos imaginamos que aquellos que lo hayan realizado se habrán guiado por el esquema original, que aparece en la página 10 del artículo teórico. No obstante, publicamos nuevamente el dibujo completo.



¡POR FIN!

Ha llegado un Nº 1

SAUCER ATTACK
es realmente
***un* JUEGO DISTINTO**



FOTOS
TOMADAS DIRECTAMENTE
DE UN MONITOR 1702
COMMODORE



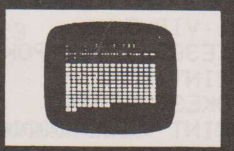
INSUPERABLE CALIDAD AUDIO-VISUAL

FERRE-MORET S.A. **más cerca de su casa**

Encontrará el juego n.° 1 de U.S.A. ¡SAUCER ATTACK! en la relación de tiendas de informática o kioscos que detallamos:

En BARCELONA
TELEUNION COMPUTER. C/ Buenos Aires, 57
NOVO-DIGIT, S. A. Aragón, 472
ELECTRONICA JOVALL. Gran de San Andrés, 129
ELECTRO AFICION. Villaroel, 104
ELECTRONICA VIVAS. Mosén Jacinto Verdaguer, 78. Santa Coloma de Gramanet
RADIO DEFOREST. Viladomat, 105
Kios Aeropuerto de Barcelona
Areas Sants Estación Sants Ferrocarriles
LaºLibrería, S. A. Avda. Sarria, 2
Lebreria Artós. Mayor de Sarria, 2
Leonor Hernández Reina Elisenda, 2
Librería Bosch Ronda. Universidad, 11
Ediciones Z. Paseo de Gracia, 19
Mallorca. Rambla cataluña, 86
Libros y revistas BRUGUERA. Rambla cataluña, 72
XAPS. Balmes, 244
ISLA DEL TESORO. Manuel Girona, 42
MANDRI. Mandri/Cerdanyola
Calvet. Pza. Núñez de Arce
Kiosco Canuda Ramblas
Librería Catalonia. Rda. S. Pedro Fte. Corte Inglés
Kiosco Vanguardia. Paseo de Gracia
Kiosco Drugstore. Paseo de Gracia
Librería Francesa. Paseo de Gracia
Kiosco LA OCA. Plaza Calvo Sotelo
Kiosco ZANON. Plaza Calvo Sotelo
Kiosco Princesa SOFIA. Frente Hotel Princesa Sofía
Kiosco Intern. Pedralbes P.° Manuel Girona
Kiosco COCA CASPE
Kiosco MARTOS. Ramblas
Kiosco Colón. Ramblas
Kiosco Carrillo. Baena Diagonal, fte. Corte Inglés
Kiosco TELE EXPRESS. P.° de Gracia
Kiosco SOLI. Ramblas Plaza Cataluña, 13
Kiosco YA. Plaza Cataluña, 13
Drugstore DAVID. Tuset
Librería Arcadia. Tuset-Pje. Arcadia
Kiosco Corte Inglés. Corte Inglés Diagonal
VIC (BARCELONA) SERVI COMPUT. C/ Moragas, 46 bjs.
VILAFRANCA DEL PENEDES. RADIO COMPUTER CENTER. Crtra. de Igualada, 21
TARRAGONA. Electrónica RIFE. C/ Ramón y Cajal, 64
GERONA. REGISCOMPTE, S. A. C/ Emilio Grahit, 17 bis
LERIDA. TELESA. Doctor Fleming, 53
NAVARRA-ANDORRA. 2 Avda. Roncesvalles, 8 PAMPLONA
ZARAGOZA. ADA COMPUTER. P.° Independencia, 24-26
BILBAO. AYMOSA. Blas de Otero, 45
SESTAO, BILBAO. C/ Vía Galindo, 2
SANTIAGO DE COMPOSTELA. TADEL. C/ Mezonzo, 15
SANTIAGO DE COMPOSTELA. HALT SOFTWARE. C/ Alférez Provisional, 2 ent. E
TORRELAVEGA, CANTABRIA. INFORMATICA SIGLO XXI. C/ San José M. Pereda, 1
BURGOS. CENTRO WELCOME. Alejandro Rodríguez Valcárcel, 9
MADRID. ABC INFORMATICA. Zurbano, 91 6-B
MADRID. KEY INFORMATICA. Embajadores, 90, tienda
LAS PALMAS DE GRAN CANARIA. Pedro González Melián. Rosario, 5, Vegueta
PALMA DE MALLORCA. DIMEL, S. A. Juan de Cremon, 4

- **TARJETA 64 K RAM. VIC 20**
- **TARJETA 40/80 COLUMNAS VIC 20**
- **TARJETA 80 COLUMNAS COMMODORE-64**
- **SLOTS DE EXPANSION (2 Y 5 TARJETAS)**
- **EXTENSO SOFTWARE 80 COLUMNAS**



- **LA TARJETA 64K RAM ES LA PERFECTA COMBINACION PARA LA 80 COL.**
  ¡CONVIERTA SU VIC-20 EN UN ORDENADOR SERIO!

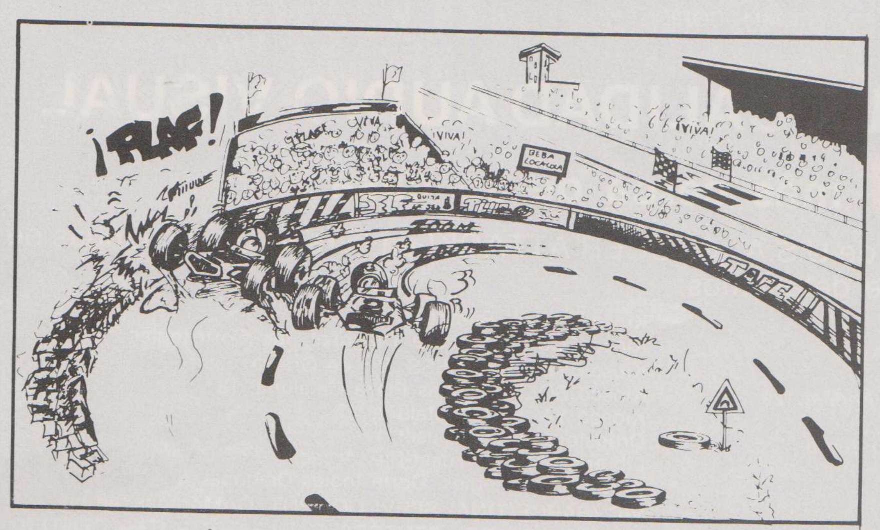
PEDIDOS A: **FERRE-MORET S.A.**

C/ TUSET, 8 08006 BARCELONA
TEL. (93) 218 02 93

Concurso

Fórmula Vic-20





Desde Barcelona nos llega FORMULA VIC, para el VIC-20 estandar. Su autor es Marcelli Subirana. Como es fácil adivinar a partir del título el programa tiene que ver con la conducción de coches, sin embargo, no se trata de ganar ninguna carrera, sino de conducir a la mayor velocidad posible por una autopista de dos vías y con bastante tráfico. El conductor dispone de dos espejos retrovisores que le permiten saber si se le acerca algún vehículo por detrás. También habrá vehículos que se dirijan hacia el conductor en dirección contraria a la que éste lleva (realmente es una autopista bastante peligrosa). El objetivo del conductor es sobrevivir tanto cuanto le sea posible y conducir su vehículo lo más aprisa que pueda. Para evitar estrellarse con los vehículos que se le vienen encima, tanto de frente como desde atrás, el conductor debe ir cambiando de carril. Para ello podrá utilizar las teclas «Z» y «X». Los coches se verán representados inicialmente en el retrovisor como dos pequeños puntos (los faros). A medida que se aproximen ira aumentado el tamaño de los puntos. Después de rebasar al conductor los coches se perderán en la lejanía. Claro está, que si el conductor no es lo bastante hábil como para cambiar a tiempo de carril no habrá nadie que pueda evitar una espectacular colisión.

La velocidad del vehículo puede modificarse mediante las teclas F1, F3, F5 y F7. Siempre hay que arrancar en primera (F1), pero una vez en marcha se puede cambiar directamente de una velocidad a otra cualquiera. F7 es la velocidad más rápida.

La estructura a grandes rasgos del programa es como sigue:

2-250	Inicialización y dibujo del coche y de la carretera.
320-330	Lectura de la tecla de arranque (F1).
400-460	Lectura del teclado.
480-990	Bucle principal.
2000-2910	Rutina de colisión.
4000-4010	Rutinas gráficas.
5000-5080	Rutinas de record y puntuación final.
6000-7030	Rutinas gráficas.
10000-20030	Rutinas de despedida y finalización.

```
2 V=1:VID=5
5 POKE36879,110:POKE36878,1:POKE650,128
10 PRINT"[CLR]"
15 POKE36869,242
20 PRINT"[CRSR] (C) \.♥UBIRANA TDZO":FORA=1TO1000:NEXT
30 FORA=1TO19:POKE7878+A,32:FORB=1TO50:NEXTB,A
46 POKE36869,240
50 PRINT"[ ]";
110 V=2:PRINT"[ ]       ";
120 PRINT"[ ]   [ ]               [ ]   [ ]";
130 PRINT"           ";
```

```
140 PRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒.__."
150 PRINT"▒▒▒▒▒▒▒▒▒▒▒▒.  :  ."
160 PRINT"▒▒▒▒▒▒▒▒▒.       ."
170 PRINT"▒▒▒▒▒▒▒.   :       ."
180 PRINT"▒▒▒▒▒▒.    :        ."
190 PRINT"▒▒▒▒▒.     :         ."
200 PRINT"▒▒▒▒.      :          ."
210 PRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒ ▙▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁"
230 PRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒"
240 PRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒"
250 PRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒"
320 GETB$:IFB$="■"THEN340
330 GOTO320
340 PRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒   PTS.    VEL.   AUTO ":AD=10:POKE36877,190:VEL=1
345 GOSUB4010:GOSUB6000
350 IFRND(0)>.5THENX=13:V=1:GO TO 370
360 X=5
370 PRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒";TAB(X)"▒▒▒"
400 FG=PEEK(203)
410 IF(FG=31ORFG=33)ANDV<>2THENPRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒";TAB(X);"":X=5:V=2:GOTO3
70
420 IF(FG=23ORFG=26)ANDV<>1THENPRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒";TAB(X)"":X=13:V=1:GOTO3
70
430 IFFG=39THENAD=13:POKE36877,180:VEL=1
440 IFFG=47THENAD=16:POKE36877,200:VEL=2
450 IFFG=55THENAD=18:POKE36877,230:VEL=3
460 IFFG=63THENAD=19:POKE36877,250:VEL=4
480 GOSUB4000:GOSUB6000
490 IFR=1THEN605
510 H=INT(RND(0)*2):IFH=0THENH=2
520 I=INT(RND(0)*2):IFI=0THENI=2
525 L=INT(RND(0)*2):IFL=0THENL=2
530 IFH=1ANDR<>1THENR=1:GOTO605
540 GOTO400
605 U=U+1:IFU<20-ADTHEN400
610 U=0:K=K+1:IFI=2ANDL=1THEN700
615 IFL=2ANDI=1THEN800
616 IFL=2ANDI=2THEN900
620 IFK=1THENPS=7755:POKEPS,43
630 IFK=2THENPOKEPS,32:PS=7842:POKEPS,108
640 IFK=3THENPOKEPS,32:PS=7929:POKEPS,35
650 IFK=4THENPOKEPS,32:GOSUB7000:IFV=2THEN2000
660 IFK=5THENGOSUB7010:R=0:PUN=PUN+10:K=0
680 GOTO400
700 IFK=1THENPS=7725:POKEPS,46:POKEPS+1,46
710 IFK=2THENPOKEPS+1,32:PS=7725:POKEPS,108:POKEPS+2,123
720 IFK=3THENPS=7725:POKEPS,81:POKEPS+2,81
725 IFK=4THENIFV=2THEN2000
730 IFK=4THENPOKEPS,32:POKEPS+2,32:GOSUB7000:PS=8016-22
740 IFK=5THENGOSUB7010:PUN=PUN+15:PS=7929:POKEPS,35
760 IFK=6THENPOKEPS,32:PS=7842:POKEPS,108
770 IFK=7THENPOKEPS,32:PS=7755:POKEPS,43
780 IFK=8THENPOKEPS,32:R=0:K=0
790 GOTO400
```

Concurso

```
800 IFK=1THENPS=7757:POKEPS,43
810 IFK=2THENPOKEPS,32:PS=7846:POKEPS,108
820 IFK=3THENPOKEPS,32:PS=7935:POKEPS,35
830 IFK=4THENPOKEPS,32:GOSUB7020:IFV=1THEN2000
845 IFK=5THENGOSUB7030:PUN=PUN+10:R=0:K=0
870 GOTO400
900 IFK=1THENPS=7725+17:POKEPS,46:POKEPS+1,46
910 IFK=2THENPOKEPS+1,32:PS=7725+17:POKEPS,108:POKEPS+2,123
920 IFK=3THENPS=7725+17:POKEPS,81:POKEPS+2,81
930 IFK=4THENPOKEPS,32:POKEPS+2,32:GOSUB7020:IFV=1THEN2000
940 IFK=5THENGOSUB7030:PUN=PUN+15:POKEPS,32:PS=7935:POKEPS,35
960 IFK=6THENPOKEPS,32:PS=7842+4:POKEPS,108
970 IFK=7THENPOKEPS,32:PS=7755+2:POKEPS,43
980 IFK=8THENPOKEPS,32:R=0:K=0
990 GOTO400
2000 POKE36877,0:FORYU=1TO100:NEXT
2010 POKE36877,220
2015 IFV=2THENPRINT"SQQQQQQQQQQQQQQ▶▶▶▶▶▶▶▶.O-":R=0
2016 IFV=1THENPRINT"SQQQQQQQQQQQQQQ▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶.O-"
2017 GETART$
2020 FORLLI=15 TO 0STEP-1
2030 POKE36878,LL
2035 FORSF=1TO2:FORTU=1TO50:NEXT
2040 POKE36879,9*LLI+SF
2050 NEXT
2060 FORAC=1TO50:NEXT:NEXT
2070 POKE36878,1:POKE36877,0
2100 V=2:K=0:R=0
2910 POKE36879,110:PRINT"J":VID=VID-1:IFVID=0THEN5000
3000 GOTO50
4000 PRINT"SQQQQQQQQQQQQQQQQQQQQQ▶▶▶▶▶▶▶▶▶▶▶▶▌    █████"VEL:RETURN
4010 PRINT"SQQQQQQQQQQQQQQQQQQQQQQ▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▌ ██"VID:RETURN
5000 IFPUN>RECTHENREC=PUN
5020 PRINT"JQQQ PUNTOS:";PUN
5030 PRINT"Q RECORD:";REC
5040 PRINT"QQQ OTRA VEZ?"
5050 GETAS$:IFAS$="N"THEN20000
5060 IFAS$<>"S"THEN5050
5070 V=2:K=0:R=0:PUN=0:VID=5
5080 GOTO50
6000 PRINT"SQQQQQQQQQQQQQQQQQQQQ▶▌      ██████"PUN
6020 RETURN
7000 PRINT"SQQQQQQQQQQQQQQQ▶▶▶▶▶▌.O. ":RETURN
7010 PRINT"SQQQQQQQQQQQQQQQ▶▶▶▶▶▌    ":RETURN
7020 PRINT"SQQQQQQQQQQQQQQQ▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▌.O. ":RETURN
7030 PRINT"SQQQQQQQQQQQQQQQ▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▌    ":RETURN
10000 IFK=4THENPOKEPS,32:GOSUB7020:IFV=1THEN2000
20000 PRINT"JQQQQQQQX     X  AAAAAA  OOOOOO";
20001 PRINT" X  X    A    A  O    O";
20002 PRINT"  XX     AAAAAA  O    O";
20003 PRINT" X  X    A    A  O    O";
20004 PRINT"X    X   A    A  OOOOOO";
20010 FORI=0TO39STEP3:POKE36876,203+I:FORJ=1TO200:NEXTJ,I
20020 FORI=36TO0STEP-3:POKE36876,203+I:FORJ=1TO150-I*5:NEXTJ,I
20030 POKE36876,0:PRINT"J"
```

Corsario

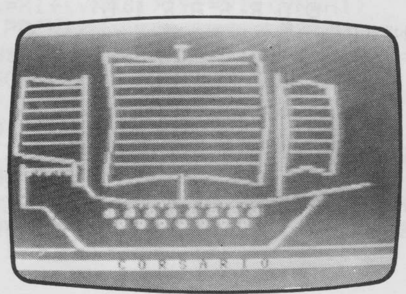
Prepárate a sostener un duro combate contra un galeón pirata, el CORSARIO, al que tendrás que perseguir y hundir, y que se encuentra navegando por los mares del sur, en las inmediaciones de tu barco. Este es el tema del juego que nos envía Miquel Clarols, lector de Commodore Magazine de Tarragona. El juego ha sido diseñado para un C-64.

Comienza pidiendo un nombre para el galeón del jugador (os aconsejamos un nombre que suene bien, como «el terror de los océanos» o «el rey de las tempestades», o algo así). A continuación aparece en la pantalla un inventario sobre el estado de vuestro barco. Elementos como el velamen, el casco y la tripulación aparecen con un porcentaje del 100%, lo que quiere decir que están intactos. A medida que se desarrolle el combate estos porcentajes irán disminuyendo indicando los daños que ha sufrido la nave.

Los galeones van a navegar guiándose por unas cartas de navegación en coordenadas cartesianas (X e Y), siendo la posición inicial del barco del jugador de (600,300). A partir de aquí comienza la navegación.

El ordenador pedirá un rumbo. Hay que contestar con un valor comprendido entre 0 y 360 grados. Entonces veremos aparecer en la pantalla varios mensajes indicándonos la velocidad del barco, su nueva posición, la posición del corsario, la distancia entre ambos y la inclinación de cada uno de ellos, expresada en minutos (esta inclinación hay que leerla como si se tratara de la aguja de los minutos de un reloj).

Cuando después de cada movimiento la posición de los barcos sea la adecuada, el ordenador permitirá disparar, en el esquema que adjuntamos se pueden ver las condiciones necesarias para poder disparar. Para disparar hay que pulsar la tecla F, mientras que si se prefiere esperar hasta estar más cerca sólo hay que pulsar RETURN.



Hay que andarse con cuidado, ya que el CORSARIO también disparará cuando su posición se lo permita, y su porcentaje de aciertos será mayor cuanto más cerca esté.

Si se decide disparar, aparecerá un mensaje pidiendo la inclinación del

Concurso

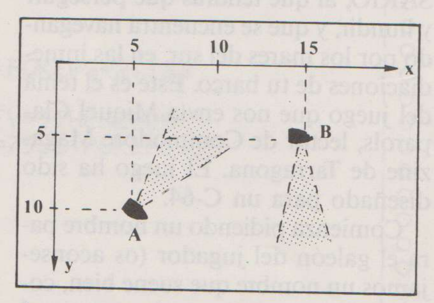
cañón. Seleccionada la inclinación, se efectúa el disparo, apareciendo en la parte superior de la pantalla dos líneas correspondientes a la altua del proyectil y a su distancia al objetivo. Si el ángulo de inclinación es muy grande o muy pequeño, el disparo irá al agua.

El programa tiene una estructura bastante complicada, pero que a grandes rasgos es como la que adjuntamos.

5-120	Dibujo de presentación.
125-150	Inicialización de variables y posiciones de los barcos.
155-165	Entrada del nombre del barco. Inicio del bucle principal.
170-200	Estado de los barcos y elección de rumbo.
205-425	Movimiento de los barcos y cálculos de posiciones.
430-500	Situación de los barcos después del movimiento.
505-560	Disparo del Corsario.
565-625	Disparo del jugador.
630-685	Movimiento de la bala de cañón y daños en el Corsario.

Para hundir al CORSARIO hay que darle justo por debajo de la línea de flotación.

ESQUEMA sobre el programa "CORSARIO"



A puede disparar (5x,10y) OH 23M
B no puede disparar (15x,5y) OH 15M

Una última advertencia, el viento siempre procede del Norte.

PREMIADO CON 5.000 PESETAS

```
5 REM "CORSARIO"
10 POKE53281,2
15 PRINT"■"
20 PRINT"
25 PRINT"
30 PRINT"
35 PRINT"
40 PRINT"
45 PRINT"
50 PRINT"
55 PRINT"
60 PRINT"
65 PRINT"
70 PRINT"
75 PRINT"
80 PRINT"
85 PRINT"
90 PRINT"
95 PRINT"               O O O O O O O O
100 PRINT"               O O O O O O O O
105 PRINT"
110 PRINT"
115 PRINT"■■            C O R S A R I O            ■"
120 FORX=1TO6000:NEXT
125 POKE53281,6
130 PRINT
135 PRINT"■"
140 X1=100:Y1=300:NN=350
145 VM=100:CM=VM:EM=VM:CJ=VM:EJ=VM:VJ=VM
150 X=600:Y=300
155 INPUT"■■INDICAR EL NOMBRE DE VUESTRO BARCO■:    ";N$
160 C$="■■CASCO   ■:":V$="■■VELAMEN■:":E$="■■TRIPULACION■:"
165 PRINT"■"
```

```
170 PRINT"CORSARIO:":PRINT E$;EM;"%":PRINTV$;VM;"%":PRINTC$;CM;"%"
175 PRINT
180 PRINT N$;":":PRINTE$;EJ;"%":PRINTV$;VJ;"%":PRINT C$;CJ"%"
185 PRINT"▩╪POS DE "N$;":X="INT (X);"/Y="INT (Y)
190 INPUT"-RUTA A SEGUIR,CAPITAN?-";CN
195 IFCN>360 THEN PRINT "-PERDON?":GOTO190
200 PRINT"-EL TIMON A "CN;" GRADOS !"
205 IFCN<40ORCN>320 THEN TP=1:GOTO210
210 FORT=1TO4STEP.2
215 IFTP=1THENY=Y+.5
220 IFCN<=180THENVN=CN*3/160:GOTO230
225 VI=CN-180:VN=(180-VI)*3/250
230 A=(COS (CN*.0174))
235 B=(SIN(CN*.0174))
240 IFCN<=90THEN X=X+(VN*B):Y=Y-(VN*A):GOTO260
245 IFCN>90AND CN<=180 THENX=X+(VN*B):Y=Y-(VN*A):GOTO260
250 IFCN>180 AND CN<=270 THENX=X+(VN*B):Y=Y-(VN*A):GOTO260
255 IFCN>270 THEN X=X+(VN*B):Y=Y-(VN*A)
260 PRINT">";:NEXT T:TP=0:PRINT
265 PRINT"-VELOCIDAD PROA :"INT(VN*5);"NUDOS"
270 SF=0:DX=INT (XI-X)
275 DY=INT(YI-Y)
280 DS=INT(SQR((DX↑2)+(DY↑2)))
285 H=DX:V=DY
290 IFDX=0 THENDX=1:IF DY=0THEN DY=1
295 A=V/H:B=ATN(A)
300 B=INT(B*57.29578)
305 IFH<0ANDV>0THEN B=90+B:GOTO320
310 IFH<0ANDV<0THEN B=90+B:GOTO320
315 B=(B+90)+180
320 IFSF=1THEN475
325 CC=INT(RND(1)+.5)
330 IFCC=>0THENCH=45
335 CH=-45
340 IFDS<NNTHENB=B+CH
350 IFB>40THEN360
355 B=B+5:GOTO350
360 IFB<320THEN375
365 B=B-5:GOTO360
370 IFDS<50THENB=B+CH
375 FORT1=1TO4STEP.2
380 PRINT">";
385 IFB<=180THENVN=B*3/160:GOTO395
390 VI=B-180:VN=(180-VI)*3/250
395 A1=(COS(B*.0174))
400 B1=(SIN(B*.0174))
405 IFB<=90THENX1=X1+(VN*B1):Y1=Y1-(VN*A1):GOTO425
410 IFB>90ANDB<=180THENX1=X1+(VN*B1):Y1=Y1-(VN*A1):GOTO425
415 IFB>180 AND B<=270THENX1=X1+(VN*B1):Y1=Y1-(VN*A1):GOTO425
420 IFB>270THENX1=X1+(VN*B1):Y1=Y1-(VN*A1)
425 NEXT T1:TP=0:PRINT
430 PRINT:PRINT N$;" ESTA EN X=" INT (X);"/Y="INT(Y);"
435 PRINT"HA HECHO RUTA A "CN;"GRADOS."
440 HM=INT(CN/6)
445 PRINT"(ESTA 0 H"HM;" MINUTOS)"
450 PRINT:X1=INT(X1):Y1=INT(Y1):PRINT"-EL CORSARIO ESTA EN X="X1;"/Y="Y1
455 PRINT"HA HECHO RUTA A"B;"GRADOS":CR=B
460 HM=INT (CR/6)
465 PRINT"(ESTA 0 H"HM;"MINUTOS)"
470 DX=INT(X1-X):DY=INT(Y1-Y):SF=1:GOTO280
```

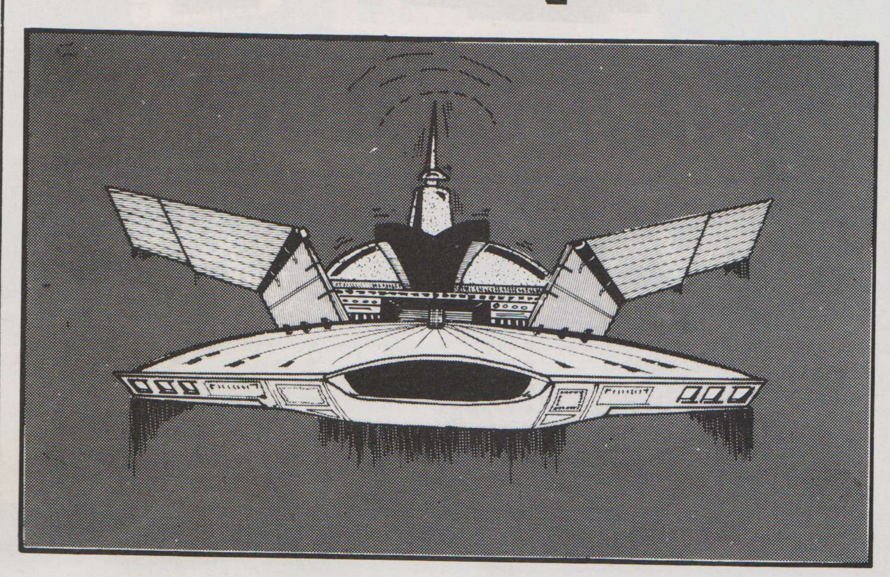
Concurso

```
475 TA=ABS(CR-B):SF=0
485 PRINT"-DISTANCIA:"DS;"METROS!"
490 IFCN<45THENCN=CN+180:IF CN>315 THEN CN=CN-180
495 IFCN=>B+45ANDCN<=B+135ORCN<=B-45 ANDCN=>B-135 THENTJ=1:GOTO505
500 TJ=0
505 PRINT:IFTA=>45ANDTA<=135THEN PRINT"TIRO ORDENADOR!";:PRINT:GOTO515
510 GOTO560
515 PT=INT(800/(DS+1))*10+5
520 TT=INT(RND(1)*99+.5)
525 FORI=1TO100STEP.8:PRINT".";:NEXTI
530 IFTT<=PTTHEN PRINT"-CARGAS TOCADAS!":GOTO540
535 PRINT"FALLO!":NN=NN-45:IFNN<100THENNN=200:GETA$:GOTO560
540 IFTT=>PT/2ANDTT<=PTTHENPRINT"VELAMEN ESTROPEADO":VJ=VJ-10:GOTO560
545 IFTT=>PT/4ANDTT<=PT/2THEN PRINT "TRIPULACION TOCADA!":EJ=EJ-10:GOTO560
550 IFTT=>PT/8ANDTT<=PT/4THENPRINT"CASCO TOCADO!":CJ=CJ-10:GOTO560
555 IFTT=>PT/20ANDTT<=PT/8THENPRINT "-EUH.NOS HUNDIMOS CAPITAN!":END
560 IFCJ=0OREJ=0ORVJ=0THEN PRINT"-NUESTRAS CARGAS PERDIDAS":END
565 IFTJ=1THEN PRINT"-PUEDE DISPARAR CAPITAN!":GOTO575
570 PRINT"NOSOTROS NO PODEMOS HACER FUEGO CAPITAN ..":GET A$:GOTO170
575 PRINT"-(F)UEGO?"
580 INPUT"";A$:IFA$<>"F"THEN 680
585 PRINT"-DISTANCIA BLANCO"DS
590 PRINT"-ANGULO CANON?"
595 INPUT"-";AN
600 G=9.81:V=50:YY=5
605 AN=AN*(3.14/180)
610 KN=COS (AN):TN=TAN(AN)
615 FORXX=1TO1000STEP3
620 YY=INT(-.5*G*(XX↑2)/(V↑2)*(KN↑2)+TN*XX)+1
625 DC=INT(DS-XX)
630 PRINT"♥"
635 PRINT"DISTANCIA AL BLANCO:"DC:PRINT
640 PRINT"ALTURA BALA:"YY-1
645 IFYY<=0THEN 655
650 NEXTXX
655 IFDC<=4AND DC=>0THENPRINT"TOCADO BAJO LA LINEA DE FLOTACION SE HUNDE":END
660 IFDC=>4 OR DC<-20 THENPRINT" PLAOUF!":GOTO680
665 IFDC=>-5AND DC<=0THEN PRINT"CASCO TOCADO !!!":CM=CM-10:GOTO680
670 IFDC=>-10ANDDC<=-5THENPRINT"TRIPULACION TOCADA!":EM=EM-10:GOTO680
675 IFDC=>-20AND DC<=-10THEN PRINT"VELAMEN TOCADO":VM=VM-10:GOTO680
680 IFCM=0OREM=0ORVM=0THEN PRINT"-EL ESTA FUERA DE COMBATE!":END
685 GETA$:XX=0:TJ=0:AN=0:KN=0:YY=5:GOTO170
```



Concurso

Modulo espacial



Sabemos que muchos de vosotros os preguntáis, antes de poneros a teclear un programa tan largo como éste, si al final mercerá la pena o no. Nuestro criterio es seleccionar los programas según un compromiso entre su calidad y su extensión, es decir, que cuanto más largo sea un programa, más exigente nos volvemos a la hora de publicarlo. MODULO ESPACIAL cumple estos requisitos. Se trata de un juego para el C-64, que nos envía Luis Llana desde Madrid, y que consiste en hacer aterrizar un módulo espacial en una base de la superficies del planeta que más os guste imaginar. Al empezar el programa, y después de las instrucciones existe la posibilidad de elegir entre cuatro bases de aterrizaje diferentes, con distinto grado de dificultad y diferente puntuación. El módulo cae hacia la superficie del planeta atraído por la gravedad. Para frenar su velocidad de caída y dirigirlo hacia la base se pueden utilizar tres cohetes propulsores, uno de ellos para dar impulso vertical, y los otros dos para controlar los desplazamientos horizontales a derecha e izquierda. Estos propulsores se pueden controlar mediante un joystick (en el port 1) o mediante las teclas «2», «CTRL» y «barra de espacios».

La fragilidad del módulo obliga a que los aterrizajes deban realizarse lo más suavemente posible. En la parte derecha de la pantalla aparecen unos indicadores del nivel de combustible y de la velocidad de caída vertical. Pues bien, sólo puede aterrizarse a salvo cuando la velocidad de caída sea inferior a 3, en cualquier otro caso el

```
10 GOTO480
20 REM **** PREPARA BUCLE PRICIPAL *****
30 AR=0:Y=0:DE=0:IZ=0:D=0:I=0:A=0:X=INT(RND(0)*145)+60:AB=1:CN=0
40 IFPR>0THENPX=33:PY=6:GOSUB2830:PRINT"▣"PR"▮▮▣ "
50 IFRE>0THENPX=33:PY=10:GOSUB2830:PRINT"▣"RE"▮▮▣ "
70 POKE53279,0:CL=0
80 REM ****** BUCLE PRINCIPAL ******
90 AB=AB+.1:Y=Y+AB-AR
100 X=X+DE-IZ
110 IFDE>0ANDD=0THENDE=DE-0.1
120 IFIZ>0ANDD=0THENIZ=IZ-0.1
130 JY=PEEK(56321)
140 IF(JY=239ORJY=254)ANDFU>0THENGOSUB320
150 IFJY=247ANDFU>0THENGOSUB370
160 IFJY=251ANDFU>0THENGOSUB420
170 X(1)=X:Y(1)=Y
180 IFY(1)<0THENY(1)=0
190 POKEV+21,A
200 IFA<>2THENPOKESO+24,7
210 IFA=2THENPOKESO+24,0
220 POKEV+2,X(1):POKEV+3,Y(1)
230 I=0:D=0:A=2
240 VE=INT((1000*(AB-AR))/100)
```

módulo se destruye al chocar con el suelo.

Otro elemento que no hay que perder de vista es el indicador de nivel del combustible, ya que cuando éste se acabe, no habrá nada que pueda evitar la destrucción del módulo, que inicia un descenso «en caída libre».

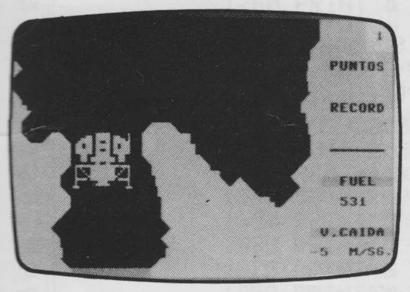
El programa, aunque largo, no tiene por qué plantear ningún problema a la hora de teclearlo, es cuestión de tomárselo con calma. Siempre se pueden eliminar las sentencias REM o las secciones de título e instrucciones, pero teniendo cuidado de no dañar elementos vitales del programa.

Adjuntamos la estructura del programa, aunque esta ya viene definida en el listado mediante un conjunto de sentencias REM.

20-70	Inicialización.
80-300	Bucle principal.
310-470	Rutinas de movimiento del módulo.
470-620	Título en caracteres gigantes.
630-770	Inicialización de sprites y música.
780-900	Instrucciones.
910-1070	Elección de la base de aterrizaje.
1080-2130	Bases de aterrizaje.
2140-2230	Mensajes del cuadro de mandos.
2240-2360	Música.
2370-2820	Rutinas de aterrizaje o explosión.
2825-2830	Rutinas de posición del módulo.
2840-3130	Datas de los diversos sprites, músicas y efectos de explosión.

Una última observación. El programa no controla las posiciones X< 0, Y >0 de los sprites 5, 6 y 7 (llamas de los cohetes propulsores) pues de hacerlo, el programa pierde velocidad y en condiciones normales no es necesario. Sin embargo, si el módulo está por encima de la parte superior de la pantalla y se hace uso de los propulsores, puede aparecer el mensaje ILLEGAL QUANTITY... Para evitar esto se pueden incluir las siguientes líneas.

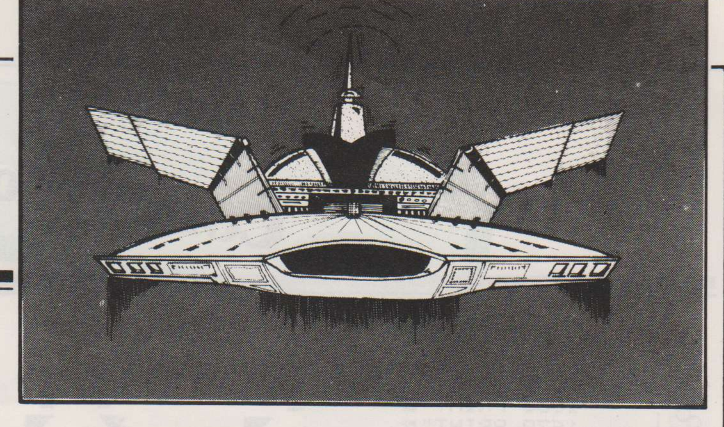
332 IFX(2) O THEN X(2) = 0
336 IFY(2) O THEN Y(2) = 0
382,386 Lo mismo con X(3) e Y(3)
442,446 Lo mismo con X(4) e Y(4)
2590 Sustituir X+10 por X(1)+10 e +1,Y+10 por +1,Y(1)+10 siendo necesario abreviar algún POKE para que quepa toda la línea.
2620 Sustituir X por X(1) así como +1,Y por +1,Y(1).
2660 Sustituir X+10 por X(1)+10 e Y+10 por Y(1)+10.




```
250 PY=22:PX=31:GOSUB2830:PRINT"▒"VE"▒ "
260 IFFU<0THENFU=0
270 PY=17:PX=33:GOSUB2830:PRINT"▒"FU"▒▒
280 CL=PEEK(53279)AND2
290 IFCL=2THENGOTO2380
300 GOTO90
310 REM ***** MOVIMIENTO ARRIBA *****
320 AR=AR+.3
330 X(2)=X+12:Y(2)=Y+38
340 POKEV+10,X(2):POKEV+11,Y(2):POKEV+21,34:A=34:FU=FU-2
350 RETURN
360 REM **** MOVIMIENTO DERECHA *****
370 DE=DE+.4:D=1
380 X(3)=X-10:Y(3)=Y+6
390 POKEV+12,X(3):POKEV+13,Y(3):POKEV+21,66:A=66:FU=FU-1
400 RETURN
410 REM **** MOVIMIENTO IZQUIERDA *****
420 IZ=IZ+.4:I=1
430 X(4)=X+34:Y(4)=Y+7
440 IFX(4)>255THENX(4)=255
450 POKEV+14,X(4):POKEV+15,Y(4):POKEV+21,130:A=130:FU=FU-1
460 RETURN
470 REM ** ROTULO 'MODULO ESPACIAL' **
480 POKE53280,7:POKE53281,7
490 PRINT"▒▒▒▒▒"
500 PRINT" ▒ ▒ ▒ ▒ ▒ ▒ ▒ ▒ ▒ ▒ ▒ ▒ ▒ ▒ ▒ ▒"
```

Concurso

```
510 PRINT" ▣█▣█▣█▣█ ▣█▣█▣█▣█▣█    ▣█ ▣█"
520 PRINT" ▣█▣█▣█▣█ ▣█▣█▣█▣█▣█    ▣█ ▣█"
530 PRINT" ▣█ ▣█▣█ ▣█▣█▣█▣█ ▣█▣█   ▣█ ▣█"
540 PRINT" ▣█  ▣█▣  █▣ █▣  █▣ █▣   █"
550 PRINT"▣▣"
560 PRINT" ▣  █▣  █▣  █▣  █▣  █▣█▣   █▣█"
570 PRINT" ▣█ █▣█  ▣█▣█▣█ ▣█▣█  ▣█▣█ ▣█▣█"
580 PRINT" ▣ █ ▣  █▣█ ▣█▣█ ▣█▣█  ▣█▣█ ▣█▣█"
590 PRINT" ▣█   ▣█▣  █▣  █▣█  ▣█▣  █▣█"
600 PRINT" ▣  █▣  █▣█  ▣█ ▣█▣  █▣█▣█ ▣█▣  █"
610 PRINT"▣▣▣   ▣GRAFICOS.... ▣LUIS LLANA ESPARCIA"
620 PRINT"▣   ▣ACCION...... ▣LUIS F. LLANA DIAZ"
630 REM **** INICIALIZA SPRITES ****
640 POKE2040,251:POKE2041,252:POKE2045,253:POKE2046,254:POKE2047,255:V=53248
650 RESTORE:FORI=16064TO16126:READA:POKEI,A:NEXT
660 FORI=16128TO16190:READA:POKEI,A:NEXT
670 FORI=16192TO16254:READA:POKEI,A:NEXT
680 FORI=16256TO16281:READA:POKEI,A:NEXT
690 FORI=16282TO16318:POKEI,0:NEXT
700 FORI=16320TO16344:READA:POKEI,A:NEXT
710 FORI=16345TO16382:POKEI,0:NEXT:V=53248:FU=700:PR=0:SO=54272
720 POKEV+28,225:POKEV+37,8:POKEV+38,2:POKEV+29,2:POKEV+23,2
730 FORI=40TO43:POKEV+I,12:NEXT:POKEV+39,7:POKEV+44,7:POKEV+45,7:POKEV+46,7
740 REM ***** INICIALIZA MUSICA ******
750 DIMAF(22),BF(22),DU(22),A1(14),B1(14),D1(I)
760 FORI=1TO22:READAF(I),BF(I),DU(I):NEXT
770 FORI=1TO14:READA1(I),B1(I),D1(I):NEXT
780 REM ******* INSTRUCIONES *******
790 PRINT"▣▣":POKE53280,6:POKE53281,6
800 PRINT"  EL MODULO HA ENTRADO EN EL CAMPO"
810 PRINT"▣DE ATRACCION DE UN LEJANO PLANETA."
820 PRINT"▣▣   ENCIENDE EL MOTOR INFERIOR O UNO"
830 PRINT"▣DE LOS DOS LATERALES DEL MODULO PARA"
840 PRINT"▣INTENTAR ATERRIZAR."
850 PRINT"▣▣   LA VELOCIDAD DE CAIDA NO PUEDE SER"
860 PRINT"▣SUPERIOR A LOS 3 METROS/SEGUNDO PARA"
870 PRINT"▣LOGRAR ATERRIZAR CORRECTAMENTE."
880 PRINT"▣▣       USA JOYSTICK O TECLADO."
890 PRINT"▣▣       ▣ PULSA BARRA ESPACIADORA "
900 GETA$:IFA$<>" "GOTO900
910 REM ******** ELEGIR BASE *********
920 PRINT"▣▣▣▣       BASES DONDE PODER ATERRIZAR:":POKE53281,6:POKE53280,6
930 PRINT"       ________________________________"
940 PRINT"▣▣  - ▣ 1 █  FACIL        = 1000 PUNTOS."
950 PRINT"▣  - ▣ 2 █  COMPLICADA  = 2000 PUNTOS."
960 PRINT"▣  - ▣ 3 █  DIFICIL     = 4000 PUNTOS."
970 PRINT"▣  - ▣ 4 █  MUY DIFICIL = 8000 PUNTOS."
980 PRINT"▣▣    TU TURNO ACABA SI SE ESTRELLA"
990 PRINT"▣    EL MODULO O SE AGOTA EL FUEL."
1000 PRINT"▣▣▣       ▣ SELECCIONA UNA BASE (1/4) █"
1010 POKE198,0:WAIT198,1
1020 GETX$
1030 IFX$="1"THENPU=1000:GOTO1090
1040 IFX$="2"THENPU=2000:GOTO1340
1050 IFX$="3"THENPU=4000:GOTO1590
1060 IFX$="4"THENPU=8000:GOTO1840
1070 GOTO1020
```

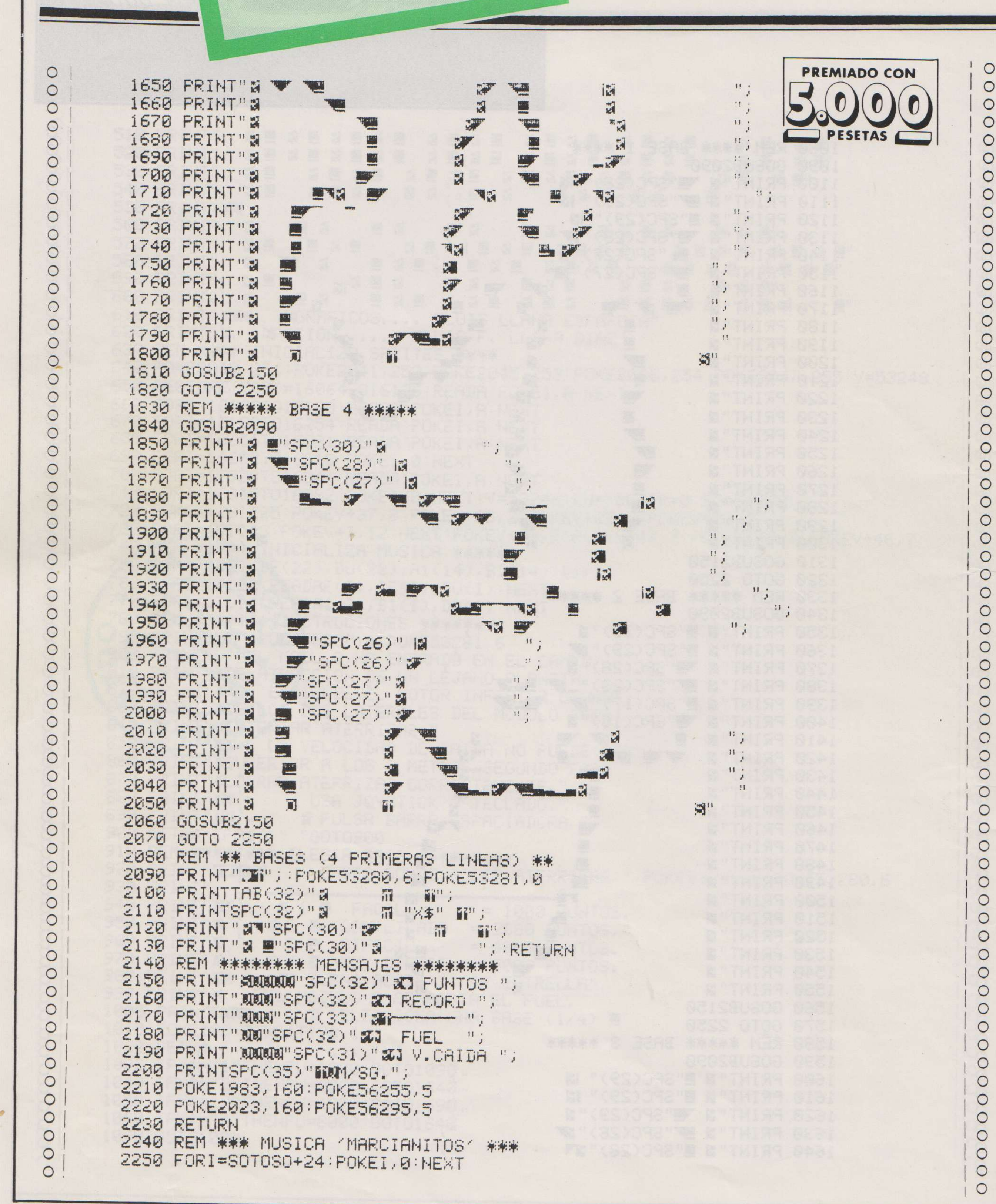


```
1080 REM **** BASE 1 ****
1090 GOSUB2090
1100 PRINT"▌ ▀"SPC(28)" ▌                    ";
1110 PRINT"▌ ▀"SPC(28)" ▌                    ";
1120 PRINT"▌ ▀"SPC(29)" ▌                    ";
1130 PRINT"▌ ▀"SPC(28)"◢                     ";
1140 PRINT"▌  ▄"SPC(27)"▌                    ";
1150 PRINT"▌  ▀"SPC(27)"▌                    ";
1160 PRINT"▌ ▀          ◢   ▀▀          ▌         ";
1170 PRINT"▌  ▀ ◢ ▀     ◢     ▀          ▌        ";
1180 PRINT"▌            ▌   ▀▄             ▌      ";
1190 PRINT"▌            ▀▌     ▀▀          ◢      ";
1200 PRINT"▌            ▀▌     ▀▀        ◢        ";
1210 PRINT"▌            ▀▌       ▀▀▌  ◢           ";
1220 PRINT"▌            ▌         ▀▀▌  ◢          ";
1230 PRINT"▌            ▌            ▀▀▌          ";
1240 PRINT"▌            ▌                         ";
1250 PRINT"▌          ▀  ◢▌                       ";
1260 PRINT"▌          ◢    ◢                      ";
1270 PRINT"▌          ▌   ◢▌                      ";
1280 PRINT"▌          ▌    ◢                      ";
1290 PRINT"▌          ▀▌    ▌                     ";
1300 PRINT"▌           ▌   ▌                  ▌";
1310 GOSUB2150
1320 GOTO 2250
1330 REM ***** BASE 2 *****
1340 GOSUB2090
1350 PRINT"▌ ▀"SPC(30)"▌                    ";
1360 PRINT"▌ ▀"SPC(29)"◢                    ";
1370 PRINT"▌ ▀"SPC(28)"▌                    ";
1380 PRINT"▌ ▀"SPC(23)"_◢ ▀◢               ";
1390 PRINT"▌ ▀"SPC(17)"◢     ▀◢            ";
1400 PRINT"▌ ▀"SPC(16)"▌                   ";
1410 PRINT"▌  ▌       ◢  ▀      ▌          ";
1420 PRINT"▌  ▀ ◢  ▀  ◢  ▀      ▀▌         ";
1430 PRINT"▌          ◢          ▌         ";
1440 PRINT"▌          ▌          ▌         ";
1450 PRINT"▌          ▌           ▌        ";
1460 PRINT"▌          ▀           ▀▌       ";
1470 PRINT"▌          ▌             ▌      ";
1480 PRINT"▌          ▌             ▌      ";
1490 PRINT"▌          ▌             ▌      ";
1500 PRINT"▌           ▀           ▌       ";
1510 PRINT"▌            ▌          ▌       ";
1520 PRINT"▌             ▌          ▌      ";
1530 PRINT"▌             ▀          ▌▌     ";
1540 PRINT"▌               ▌       ◢       ";
1550 PRINT"▌               ▌       ▌      ▌";
1560 GOSUB2150
1570 GOTO 2250
1580 REM ***** BASE 3 *****
1590 GOSUB2090
1600 PRINT"▌ ▀"SPC(29)" ▌                  ";
1610 PRINT"▌ ▀"SPC(29)" ▌                  ";
1620 PRINT"▌ ▀"SPC(29)"▌                   ";
1630 PRINT"▌ ▀"SPC(26)"◢                   ";
1640 PRINT"▌ ▀"SPC(28)"◢                   ";
```

PROGRAMA ★ commodore Magazine ★ COMPROBADO

Concurso

PREMIADO CON **5.000** PESETAS



```
1650 PRINT"                                   ";
1660 PRINT"                                   ";
1670 PRINT"                                   ";
1680 PRINT"                                   ";
1690 PRINT"                                   ";
1700 PRINT"                                   ";
1710 PRINT"                                   ";
1720 PRINT"                                   ";
1730 PRINT"                                   ";
1740 PRINT"                                   ";
1750 PRINT"                                   ";
1760 PRINT"                                   ";
1770 PRINT"                                   ";
1780 PRINT"                                   ";
1790 PRINT"                                   ";
1800 PRINT"                                   "
1810 GOSUB2150
1820 GOTO 2250
1830 REM ***** BASE 4 *****
1840 GOSUB2090
1850 PRINT" "SPC(30)" ";
1860 PRINT" "SPC(28)" ";
1870 PRINT" "SPC(27)" ";
1880 PRINT"                                   ";
1890 PRINT"                                   ";
1900 PRINT"                                   ";
1910 PRINT"                                   ";
1920 PRINT"                                   ";
1930 PRINT"                                   ";
1940 PRINT"                                   ";
1950 PRINT"                                   ";
1960 PRINT" "SPC(26)" ";
1970 PRINT" "SPC(26)" ";
1980 PRINT" "SPC(27)" ";
1990 PRINT" "SPC(27)" ";
2000 PRINT" "SPC(27)" ";
2010 PRINT"                                   ";
2020 PRINT"                                   ";
2030 PRINT"                                   ";
2040 PRINT"                                   ";
2050 PRINT"                                   "
2060 GOSUB2150
2070 GOTO 2250
2080 REM ** BASES (4 PRIMERAS LINEAS) **
2090 PRINT"";:POKE53280,6:POKE53281,0
2100 PRINTTAB(32)"       ";
2110 PRINTSPC(32)"   "X$"  ";
2120 PRINT" "SPC(30)"      ";
2130 PRINT" "SPC(30)"       ";:RETURN
2140 REM ******** MENSAJES ********
2150 PRINT""SPC(32)" PUNTOS ";
2160 PRINT""SPC(32)" RECORD ";
2170 PRINT""SPC(33)" ───────";
2180 PRINT""SPC(32)"  FUEL  ";
2190 PRINT""SPC(31)" V.CAIDA ";
2200 PRINTSPC(35)"M/SG.";
2210 POKE1983,160:POKE56255,5
2220 POKE2023,160:POKE56295,5
2230 RETURN
2240 REM *** MUSICA 'MARCIANITOS' ***
2250 FORI=SOTOSO+24:POKEI,0:NEXT
```

```
2260 POKESO+3,8
2270 POKESO+5,41:POKESO+6,89
2280 POKESO+14,117:POKESO+18,16
2290 POKESO+24,143
2300 FORI=1TO14
2310 POKESO+4,65
2320 POKESO,B1(I):POKESO+1,A1(I)
2330 FORT=1TOD1(I):NEXT:POKESO+4,64
2340 NEXT
2350 POKESO+5,15:POKESO+6,245:POKESO+1,25:POKESO,177:POKESO+24,0:POKESO+4,129
2360 GOTO30
2370 REM ***** HAS ATERRIZADO O NO *****
2380 IFVE<=3ANDY(1)>=201THENCN=1
2390 IFCN=0THEN2580
2400 REM ****** HAS ATERRIZADO *******
2410 PR=PR+PU:PX=33:PY=6:GOSUB2830:PRINT"■"PR"■■ "
2420 IFPR>RETHENRE=PR
2430 PX=33:PY=10:GOSUB2830:PRINT"■"RE"■■ "
2440 POKEV+21,2
2460 FORI=SOTOSO+24:POKEI,0:NEXT
2470 POKESO+3,8
2480 POKESO+5,41:POKESO+6,89
2490 POKESO+14,117:POKESO+18,16
2500 POKESO+24,143
2510 FORI=1TO22
2520 POKESO+4,65
```

PROGRAMA commodore Magazine COMPROBADO

Concurso

```
2530 POKESO+1,AF(I):POKESO,BF(I)
2540 FORT=1TODU(I)*9:NEXT:POKESO+4,64
2550 NEXT
2560 POKEV+21,0:POKEV+2,0:POKEV+3,0
2570 GOTO920
2575 REM **** TE HAS ESTRELLADO ****
2580 FU=700:POKESO+24,15:POKEV+23,2:POKEV+29,2
2590 POKEV,X+10:POKEV+1,Y+10:POKEV+2,X(1):POKEV+3,Y(1):POKESO+1,4:POKESO,208
2600 POKEV+21,3:POKESO+24,15:POKEV+23,2:POKEV+29,2
2610 FORI=1TO700:NEXT
2620 POKEV,X:POKEV+1,Y:POKEV+23,3:POKEV+29,3
2630 FORI=1TO600:NEXT
2640 POKEV+21,1
2650 FORI=1TO400:NEXT
2660 POKEV,X+10:POKEV+1,Y+10:POKEV+21,1:POKEV+23,2:POKEV+29,2:POKESO+24,7
2670 FORI=1TO400:NEXT
2680 POKEV+21,0:POKESO+24,0
2690 POKE53280,2:PRINT"██";:FORI=1TO24
2700 PRINT"█                                                  ";:FORT=1TO50:NEXT:NEXT
2710 POKE53281,2:PRINT"J"
2720 POKE53280,6
2730 PRINT"██████            SE ACABO TU TURNO."
2740 PRINT"           ________________________"
2750 PRINT"████        PUNTOS CONSEGUIDOS:  ██"PR"██ "
2760 PRINT"███       RECORD:  ██"RE"██ "
2770 PRINT"██████              ██ OTRA VEZ ? (S/N) "
2780 IFPR>RETHENRE=PR
2790 GETA$
2600 IFA$="S"THENPR=0:GOTO920
2810 IFA$="N"THENEND
2820 GOTO2790
2825 REM ***** RUTINA POS. X,Y ******
2830 POKE783,0:POKE782,PX:POKE781,PY:SYS65520:RETURN
2840 REM***** EXPLOSION ******
2850 DATA64,0,4,64,4,20,80,4,84,84,20,85,21,85,80,21,85,80,85,170,84,22,169,80
2860 DATA22,169,85,86,170,84,85,170,80,21,170,84,22,169,84,22,169,85
2670 DATA6,170,84,6,170,148,21,165,149,85,85,84,21,5,68,5,1,69,4,1,0
2880 REM****** MODULO **********
2890 DATA0,126,2,7,126,226,31,126,250,63,102,254,119,126,222,119,126,222
2900 DATA127,102,254,127,126,254,28,126,56,29,255,184,29,255,184
2910 DATA0,102,0,127,255,254,81,255,138,73,255,146,69,255,162
2920 DATA67,255,194,125,255,190,64,60,2,64,126,2,224,0,7
2930 REM***** TOBERA INFERIOR ******
2340 DATA3,105,192,3,105,192,3,105,192,15,105,240,13,105,112,13,105,112
2950 DATA13,105,112,13,105,112,13,105,112,15,105,240,3,105,192,3,105,192
2960 DATA3,105,192,3,85,192,3,85,192,3,215,192,0,215,192,0,215,192,0,12,0
2970 DATA0,0,0,0,0,0
2980 REM**** TOBERA IZQUIERDA *****
2990 DATA0,0,0,21,64,0,213,80,0,86,160,0,106,160,0,86,160,0
3000 DATA213,80,0,21,64,0,0,0,0
3010 REM**** TOBERA DERECHA ******
3020 DATA0,0,0,1,84,0,5,87,0,10,149,0,10,169,0,10,149,0,5,87,0,1,84,0
3030 REM *** MUSICA "NUEVO MUNDO" ***
3040 DATA34,75,40,34,75,20,28,214,20,25,177,40
3050 DATA34,75,30,43,52,10,51,92,20,51,92,20,51,92,60,51,92,20
3060 DATA57,172,5,51,97,20,45,198,20,43,52,20
3070 DATA51,97,40,45,198,10,51,97,10,45,198,10
3080 DATA38,126,10,32,94,10,28,214,10,25,177,40
3090 REM *** MUSICA "MARCIANITOS" ***
3100 DATA12,216,100,17,37,300,21,154,60,21,154,60
3110 DATA17,37,60,21,154,100,25,177,900,0,0,100
3120 DATA25,177,100,34,75,300,43,52,60,34,75,60
3130 DATA43,52,100,51,97,900
```

Halterofilia

¿Qué tal andan nuestros lectores de fuerza? Esperamos que bien pues os vamos a proponer un programa nada menos que de levantamiento de pesas, para ver que tal andáis de forzudos. Además se trata de que intenteis batir todos los records, nada de cansarse enseguida y abandonar.

La idea de este programita es de Guillermo Abella, que nos lo envía desde Castellón. Está diseñado para funcionar en un Commodore 64 (y por supuesto funciona).

El desarrollo del programa es como sigue. Después de escribir RUN aparecerá el título del programa y dos columnas con números del 10 al 90, que representan los kilos por los que se puede empezar. Es aconsejable no confiarse demasiado y empezar con 10 Kg. para no llevarse sorpresas.

Después de seleccionar el nivel de comienzo aparecerá en escena el levantador y esperará hasta que se pulse cualquier tecla para comenzar el levantamiento.

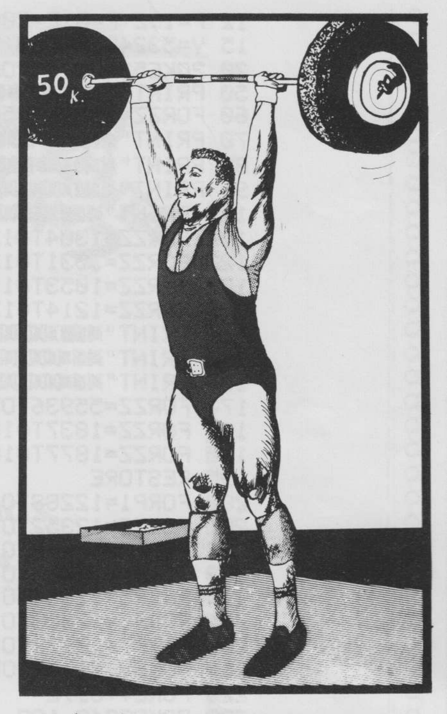
Una vez que da comienzo el levantamiento todo depende del jugador. Para hacer que las pesas se eleven, este debe pulsar repetidamente y lo más deprisa que pueda las telcas «B» y «V».

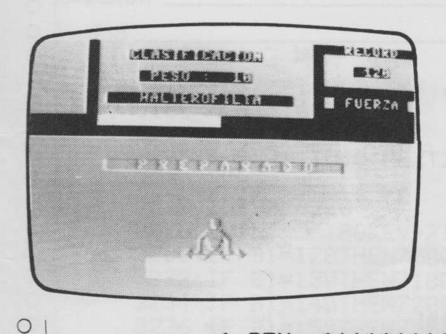
En la parte superior de la pantalla aparecen dos indicadores, uno, indica la energía del jugador y viene a ser una limitación de tiempo. El otro indica la fuerza que está aplicando el levantador. Cuanto más rápidamente se pulsen «B» y «V» mayor será la fuerza y más alto subirán las pesas.

Si se consigue que las pesas lleguen arriba, el levantador manifestará su alegría saltando, y después de reposar un poco tendrá que pasar al peso superior (10 kilitos más). A los 120 Kg. se disputa la medalla de bronce y luego las de plata y oro. A estos niveles ya se puede subir al pódium al terminar la partida.

Guillermo nos comenta que el record entre sus amigos es de 120 Kg. (medalla de bronce). Así que ánimo, que las de plata y oro están todavía por disputar.

Y una última cosa, ¡qué la fuerza os acompañe!





ESTRUCTURA DEL PROGRAMA

Líneas	Descripción	Líneas	Descripción
11-350	Diseño del dibujo y espera hasta que se pulse una tecla.	3000-3240	Salto a rutinas de las medallas.
400-700	Bucle Principal, lectura del teclado, incremento de fuerza (F). Disminución de energía.	5000-5500	Presentación y selección del Peso inicial.
1000-1040	Fin de juego y vuelta a empezar.	6000-7510	Obtención de medallas.
		8000-9200	Entrega de medallas y subida al Podium.
		10000-10330	Datas.

```
1 REM ******************************
2 REM **                          **
3 REM **      HALTEROFILIA        **
4 REM **                          **
5 REM **          POR             **
6 REM **                          **
7 REM **   GUILLERMO ABELLA       **
8 REM ******************************
9 A1=120
10 GOTO 5000
11 PRINT"[controles]CLASIFICACION"
```

Concurso

PREMIADO CON 5.000 PESETAS

```
12 P=192:F=0:E=200:Z=1
15 V=53248:X=158:Y=170
30 POKE53280,6:POKE53281,5
50 PRINT"… PESO : ";B1
60 FORZZ=55296TO56295:POKEZZ,0:NEXT
70 PRINT"… RECORD "
80 PRINT"…"A1
90 PRINT"… FUERZA "
100 PRINT"… "
110 FORZZ=1304TO1383:POKEZZ,160:NEXT
120 FORZZ=1031TO1311STEP40:POKEZZ,160:NEXT
130 FORZZ=1053TO1333STEP40:POKEZZ,160:NEXT
140 FORZZ=1214TO1223:POKEZZ,160:NEXT
150 PRINT"… HALTEROFILIA "
160 PRINT"…ENERGIA:"
165 PRINT"…";E
170 FORZZ=55936TO56295:POKEZZ,7:NEXT
180 FORZZ=1837TO1849:POKEZZ,160:NEXT
190 FORZZ=1877TO1889:POKEZZ,160:NEXT
195 RESTORE
200 FORP1=12288TO12350:READK1:POKEP1,K1:NEXT
201 FORP2=12352TO12414:READK2:POKEP2,K2:NEXT
202 FORP3=12416TO12478:READK3:POKEP3,K3:NEXT
203 FORP4=12480TO12542:READK4:POKEP4,K4:NEXT
204 FORP5=12544TO12606:READK5:POKEP5,K5:NEXT
205 FORP6=12608TO12670:READK6:POKEP6,K6:NEXT
206 FORP7=12672TO12734:READK7:POKEP7,K7:NEXT
210 POKEV+21,1:POKEV,X:POKEV+1,Y
220 POKEV+39,2
230 POKE2040,192
240 POKEV+23,1:POKEV+29,1
300 S=54272
305 FORL=0TO24:POKES+L,0:NEXT
310 POKES+14,5:POKES+18,16:POKES+3,1
315 POKES+24,143:POKES+6,240:POKES+4,65
320 FR=5389
325 FORT=1TO10
330 FQ=FR+PEEK(S+27)*3.5
335 HF=INT(FQ/256):LF=FQ-HF*256
340 POKES+0,LF:POKES+1,HF
345 NEXT
350 POKES+24,0
360 PRINT"… P R E P A R A D O "
370 FORZZ=1TO30:GETA$:NEXT
375 GETA$:IF A$=""THEN375
380 PRINT"… "
400 FORBB=1TOZ:GETA$:NEXT
405 GETA$:IF A$="V"ORA$="B"THENF=F+20:GOTO 420
410 F=F-5
420 PRINT"… "
425 IF F<0THENF=0
430 PRINT"…";F
438 IF F>1200 THENZ=10:GOTO 550
439 IF F>1100 THENZ=9:GOTO 550
440 IF F>1000 THENZ=8:GOTO 550
445 IF F>900 THENZ=7:GOTO 550
450 IF F>800 THENZ=6:GOTO 550
455 IF F>700 THENZ=5:GOTO 550
460 IF F>600 THENZ=4:GOTO 550
470 IF F>400 THENZ=3:GOTO 550
480 IF F>200 THENZ=2:GOTO 550
```

```
550 IF F>L1THENP=197:GOTO 570
556 IF F>L2THENP=196:GOTO 570
557 IF F>L3THENP=195:GOTO 570
558 IF F>L4THENP=194:GOTO 570
559 IF F>L5THENP=193:GOTO 570
560 IF F<L5THENP=192:GOTO 570
561 IF F<L4THENP=193:GOTO 570
562 IF F<L3THENP=194:GOTO 570
563 IF F<L2THENP=195:GOTO 570
564 IF F<L1THENP=196:GOTO 570
570 POKE2040,P
575 IF P=197 THEN 3000
583 E=E-1
587 IF E<0THEN1000
588 PRINT"[illegible]              "
590 PRINT"[illegible]";E
700 GOTO 400
1000 GOTO 7500
1005 PRINT"[illegible] J U E G O   T E R M I N A D O "
1010 FORZZ=1TO6000:NEXT
1020 PRINT"[illegible]":POKEV+21,0
1040 GOTO 10
3000 GOSUB 4000
3010 FORZZ=1TO2500:NEXT
3070 L1=L1+50:L2=L2+40:L3=L3+35:L4=L4+25:L5=L5+15:B1=B1+10
3030 FORP=196TO192STEP-1
3100 POKE2040,P
3110 FORZZ=1TO250:NEXTZZ
3120 NEXT
3130 POKE2040,198
3135 GOSUB 4000
3140 FORY=170TO140STEP-1
3150 POKEV+1,Y
3155 FORZZ=1TO10:NEXTZZ
3160 NEXT
3165 IF B1>A1-1+10THEN6000
3170 FORZZ=1TO50:NEXTZZ
3180 FORY=140TO170
3190 POKEV+1,Y
3195 FORZZ=1TO10:NEXTZZ
3200 NEXT
3210 GOSUB 4000
3220 PRINT"[illegible]":POKEV+21,0
3230 IF B1=120THEN7000
3232 IF B1=130THEN7100
3234 IF B1=140THEN7200
3236 IF B1=150THEN7300
3238 IF B1=160THEN7400
3240 GOTO 11
4000 VV=54272
4005 FORL=0TO24:POKEVV+L,0:NEXT
4010 POKEVV+24,15:POKEVV+6,0
4020 MM=1
4030 POKEVV+1,24:POKEVV+5,138
4040 POKEVV+4,129:FORNN=1TO1500-75*MM:NEXTNN
4050 POKEVV+4,0
4100 RETURN
5000 PRINT"[illegible]":POKEV+21,0
```

Concurso

```
5010 POKE53280,6:POKE53281,5
5020 PRINT"████████████ H A L T E R O F I L I A "
5025 PRINT"█████10████20████30████40████50"
5030 PRINT"███KILOS██████████████████████████████KILOS"
5035 PRINT"██████████60████70████80████90"
5040 PRINT"██████████████████████████  K I L O S "
5050 GETA$
5060 IF A$="1"THENB1=10:GOTO 5100
5061 IF A$="2"THENB1=20:GOTO 5100
5062 IF A$="3"THENB1=30:GOTO 5100
5063 IF A$="4"THENB1=40:GOTO 5100
5064 IF A$="5"THENB1=50:GOTO 5100
5065 IF A$="6"THENB1=60:GOTO 5100
5070 IF A$="7"THENB1=70:GOTO 5100
5080 IF A$="8"THENB1=80:GOTO 5100
5090 IF A$="9"THENB1=90:GOTO 5100
5092 IF A$=""THEN 5050
5095 FORZZ=1TO2:GETA$:NEXT
5097 GOTO 5050
5100 PRINT"██████████████████████████████████";B1
5105 FORZZ=1TO1000:NEXT
5110 L5=35:L4=75:L3=115:L2=160:L1=200
5120 FORQS=1TOB1/10
5125 L5=L5+15:L4=L4+25:L3=L3+35:L2=L2+40:L1=L1+50
5130 NEXT
5500 GOTO 11
6000 A1=B1
6010 GOSUB 4000
6100 GOTO 3180
7000 PRINT"████████████████ B R O N C E "
7010 GOTO 12
7100 PRINT"████████████████  P L A T A  "
7120 GOTO 12
7200 PRINT"████████████████  O   R   O  "
7220 GOTO 12
7300 PRINT"████████████████CAMPEON OLIMPICO"
7320 GOTO 12
7400 PRINT"████████████████CAMPEON DEL MUNDO"
7420 GOTO 12
7500 IF B1>120 THEN8000
7510 GOTO 1005
8000 POKEV+21,0
8010 FORZZ=1384TO2023:POKEZZ,160:NEXT
8020 FORZZ=55656TO56295:POKEZZ,5:NEXT
8030 IF B1=130THEN8060
8040 IF B1=140THEN8070
8045 IF B1=150THEN8080
8047 IF B1=160THEN8090
8050 IF B1=170THEN9000
8060 X=168:Y=165
8064 PRINT"████████████████████████████████████ M E D A L L A  D E  B R O N C E "
8066 GOTO 9100
8070 X=128:Y=157
8074 PRINT"████████████████████████████████████ M E D A L L A  D E  P L A T A "
8076 GOTO 9100
8080 X=88:Y=141
8084 PRINT"████████████████████████████████████ M E D A L L A  D E  O R O "
8086 GOTO 9100
```

PROGRAMA commodore Magazine COMPROBADO

No.2

PRIMERA REVISTA EN CASSETTE

DICIEMBRE/ENERO 84/85

64 commodore TAPE COMPUTING

REVISTA INDEPENDIENTE EN CASSETTE PARA USUARIOS DEL COMMODORE 64

875 ptas.

Utilidades
Como hacer Scroller
Sprites en alta resolución

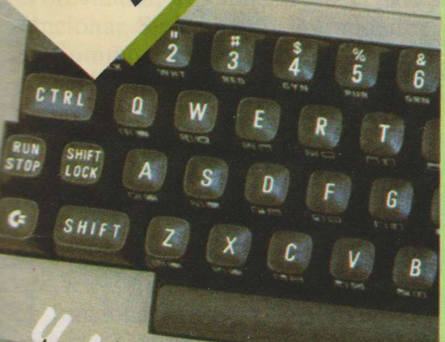
Juegos
Punkman
Mine zone

Aplicaciones
Fichero de direcciones
Disfrute dibujando

64 commodore TAPE COMPUTING

ALGO SOBRE SCROLLER
PUNKMAN
SPRITES EN ALTA RESOLUCION
MUSICAL SIMON
GHOSTY
GRAFICOS EN ALTA RESOLUCION
MATEMATICA BASICA
MINE ZONE
FICHERO DE DIRECCIONES
MUSICA POR INTERRUPCIONES
MOVING MAZE

Educativos
Simón musical
Matematica basica



¡Ya está a la venta!
Cómprela en su quiosco habitual o solicítela a:
INFODIS, S.A.
Bravo Murillo, 377-5.º-A
28020 MADRID
enviando el cupón que encontrará en esta revista



Concurso

```
8090 X=88:Y=141
8094 PRINT"[illegible] C A M P E O N  O L I M P I C O "
8096 GOTO 9100
9000 X=88:Y=141
9010 PRINT"[illegible] C A M P E O N  D E L  M U N D O "
9020 GOTO 9100
9100 PRINT"[illegible]                    "
9110 PRINT"[illegible]               [illegible]3[illegible] "
9120 PRINT"[illegible]        [illegible]2[illegible] "
9130 PRINT"[illegible]          "
9140 PRINT"[illegible] [illegible]1[illegible] "
9150 FORZZ=1TO1000:NEXT
9155 POKEV+23,0:POKEV+29,0
9160 POKEV,X:POKEV+1,Y:POKE2040,198:POKEV+21,1
9165 GOSUB 4000
9170 FORZZ=1TO3000:NEXT
9200 GOTO 5000
10000 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,28,0
10010 DATA0,28,0,0,28,0,0,8,0,0,126,0,0,255,0,1,255,128
10020 DATA3,126,192,2,126,64,2,60,64,2,60,64,34,24,68,102,126,102
10030 DATA255,255,255,96,195,6,35,129,196
10050 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
10060 DATA0,28,0,0,28,0,0,28,0,0,8,0,63,255,252,49,255,140
10070 DATA24,126,24,40,126,20,108,60,54,255,255,255,96,24,6,32,126,4
10060 DATA0,231,0,0,195,0,3,129,192
10100 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
10110 DATA0,28,0,0,28,0,0,28,0,0,8,0,33,255,132,97,255,134
10120 DATA255,255,255,121,126,158,45,60,180,7,60,224,0,24,0,0,126,0
10130 DATA0,231,0,0,195,0,3,129,192
10150 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
10160 DATA0,28,0,32,28,4,96,28,6,255,255,255,121,255,158,49,255,140
10170 DATA25,126,152,13,126,176,7,60,224,0,60,0,0,24,0,0,126,0
10180 DATA0,231,0,0,195,0,3,129,192
10200 DATA0,0,0,0,0,0,0,0,0,0,32,0,4,96,0,6
10210 DATA255,255,255,99,28,198,38,28,100,12,8,48,15,255,250,1,255,128
10220 DATA0,126,0,0,126,0,0,60,0,0,60,0,0,24,0,0,126,0
10230 DATA0,231,0,0,195,0,3,129,192
10250 DATA32,0,4,96,0,6,255,255,255,99,0,198,33,0,132,1,0,128
10260 DATA1,28,128,1,28,128,1,28,128,1,8,128,1,255,128,1,255,128
10270 DATA0,126,0,0,126,0,0,60,0,0,60,0,0,24,0,0,126,0
10280 DATA0,231,0,0,195,0,3,129,192
10300 DATA0,0,0,0,0,0,0,0,3,0,192,1,0,128,1,0,128
10310 DATA1,28,128,1,28,128,1,28,128,1,8,128,1,255,128,1,255,128
10320 DATA0,126,0,0,126,0,0,60,0,0,60,0,0,24,0,0,126,0
10330 DATA0,231,0,0,195,0,3,129,192
```

PROGRAMA commodore Magazine COMPROBADO

Cómo guarda el diskette la información

El medio más comunmente utilizado por los ordenadores personales para almacenar información es el magnético. Existen dos soportes físicos principales: la cinta y el *diskette*. Por supuesto, para cada uno de ellos hay distintos formatos. Para el **VIC-20** y el **C-64** se recurre a la cinta de *cassette*, del mismo tipo de las utilizadas en audio, aunque el medio magnético puede ser de otro tipo más adecuado para grabar señales digitales, y al *diskette*, que es un circulo de material plástico recubierto por el medio magnético. El acceso a la información es mucho más rápido en el *diskette*, pues la cabeza de lectura/grabación puede desplazarse por toda la superficie circular de modo aleatorio. Por el contrario, el acceso a la información contenida en cinta es secuencial, es decir, la cabeza debe leer toda la información que se encuentra almacenada entre su posición actual y la posición que ocupa lo que buscamos.

Los dos modelos de unidad de *diskette* más divulgados para los modelos mencionados de unidad son los **1540 y 1541.** Normalmente las unidades de *diskette* son controladas desde el ordenador, dependiendo de él para funcionar. Sin embargo, las unidades de **Commodore** son autónomas, disponen de un microprocesador propio incorporado, una pequeña cantidad de memoria RAM y una memoria ROM que contiene el sistema operativo.

El sistema operativo es un conjunto de programas en código máquina que se encargan de gestionar el correcto funcionamiento de la unidad de *diskettes* en combinación con el ordenador. Su nombre es DOS, *Disk Operating System* (sistema operativo para disco).

Comencemos viendo de qué manera se almacena la información en un *diskette* cualquiera. Obviamente no se dispone de manera aleatoria en la superficie del *diskette*, sino que debe ir colocada en un determinado orden para que el DOS sepa donde buscarla en cualquier momento.

El *diskette* almacena la información en forma de bytes, tal como lo hace la memoria RAM central del ordenador, y la deposita sobre una serie de pistas invisibles que forman circunferencias concéntricas en el disco.

El número de pistas en que se divide al *diskette* es de 35. Puede parecer

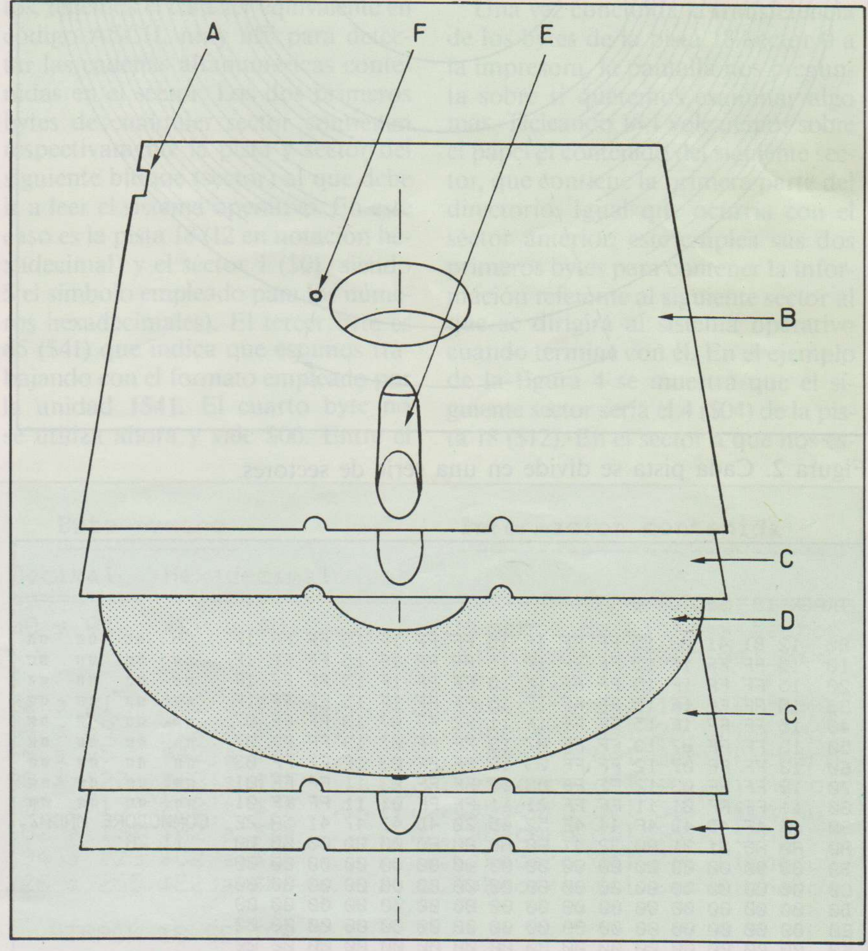


Figura 1. Composición del *diskette*. A. Ventana de escritura. Si no existe, el disco está protegido. B. Bolsa protectora. C. Tela limpiadora del disco mientras gira. D. Disco magnético. E. Ventana por la que se leen y graban los datos. F. Agujero para el sincronismo.

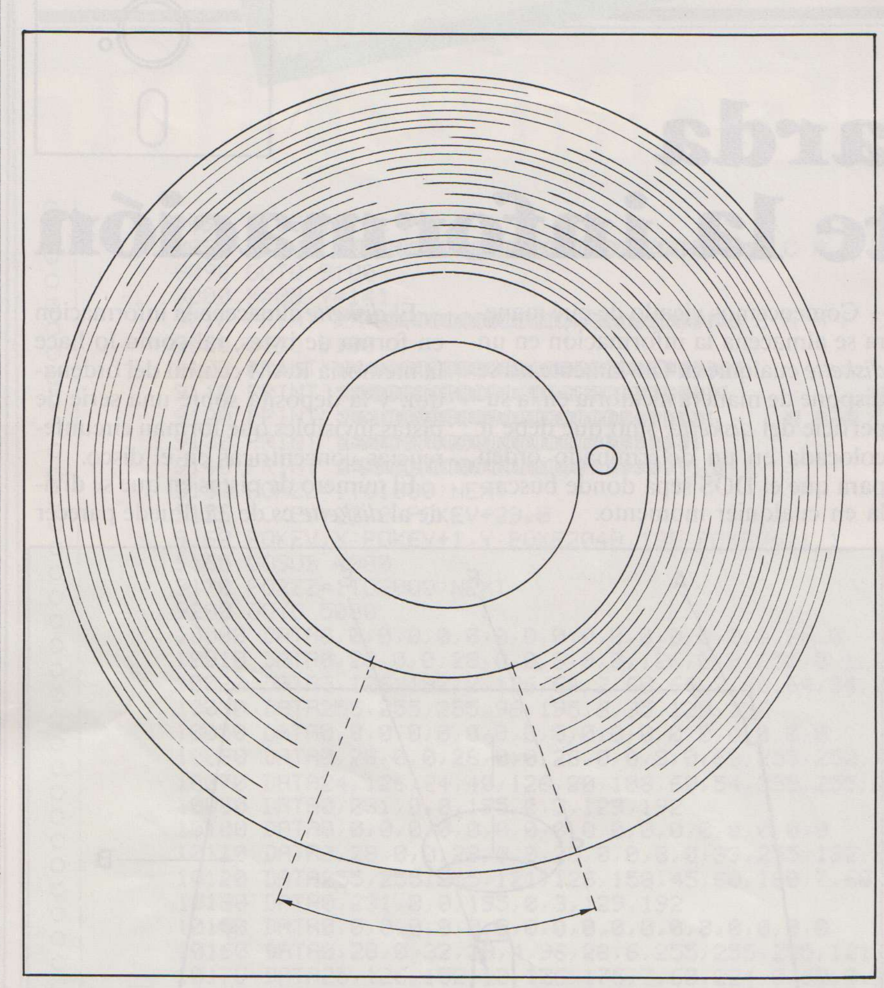


Figura 2. Cada pista se divide en una serie de sectores.

```
TRACK 18  SECTOR 0

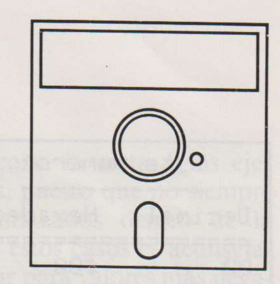
00 :12 01 41 00 15 FF FF 1F 15 FF FF 1F 15 FF FF 1F :  A  ππ  ππ  ππ
10 :15 FF FF 1F 15 FF FF 1F 15 FF FF 1F 15 FF FF 1F : ππ  ππ  ππ  ππ
20 :15 FF FF 1F 15 FF FF 1F 15 FF FF 1F 15 FF FF 1F : ππ  ππ  ππ  ππ
30 :15 FF FF 1F 15 FF FF 1F 15 FF FF 1F 15 FF FF 1F : ππ  ππ  ππ  ππ
40 :15 FF FF 1F 15 FF FF 1F 11 FC FF 07 13 FF FF 07 : ππ  ππ  ▪π  ππ
50 :13 FF FF 07 13 FF FF 07 13 FF FF 07 13 FF FF 07 : ππ  ππ  ππ  ππ
60 :13 FF FF 07 12 FF FF 03 12 FF FF 03 12 FF FF 03 : ππ  ππ  ππ  ππ
70 :12 FF FF 03 12 FF FF 03 12 FF FF 03 11 FF FF 01 : ππ  ππ  ππ  ππ
80 :11 FF FF 01 11 FF FF 01 11 FF FF 01 11 FF FF 01 : ππ  ππ  ππ  ππ
90 :43 4F 4D 4D 4F 44 4F 52 45 20 4D 41 47 41 5A 2E :COMMODORE MAGAZ.
A0 :A0 A0 31 31 A0 32 41 A0 A0 A0 A0 00 00 00 00 00 :  11 2A
B0 :00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :
C0 :00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :
D0 :00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :
E0 :00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :
F0 :00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 :
```

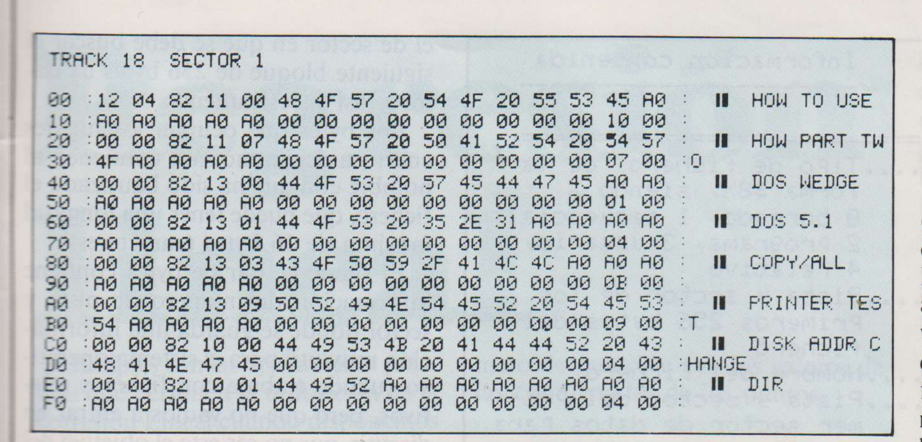
Figura 3. Volcado en impresora de la pista 18/sector 0.

extraño esto último que acabamos de decir, pero imaginemos un *diskette* recién comprado. En él no hay más que la capa de material magnético, formado por microscópicas partículas, no existe en él nada que pueda parecerse a las pistas mencionadas. Es el DOS quien desplaza la cabeza grabadora de la unidad de *diskettes* encima de un punto determinado de la superficie, dejándola inmovil. A su vez el *diskette* gira por medio de otro motor y esto es lo que produce que la información grabada sobre el disco lo sea en disposición circular. Posteriormente el sistema operativo vuelve a desplazar la cabeza a otro punto próximo y se repite el proceso. La información que va depositando el DOS sobre cada una de las pistas informará posteriormente a la cabeza de lectura sobre el lugar en que está situada en cada momento, pues contiene los datos referentes a la pista y sector afectado. Aparece aquí un concepto que no habíamos mencionado antes: sector. Para mayor flexibilidad en el manejo se divide a cada pista en una serie de sectores que varian en cantidad según el número de la pista. La pista número 1 es la más exterior y por tanto su longitud es la mayor, siendo dividida por el DOS en 21 bloques o sectores. Esta cifra es la misma hasta la pista 17. Entre la 18 y la 24 el número de sectores baja a 19, siendo de 18 desde la 25 a la 30 y 17 entre la 31 y la 35. Este proceso inicial por el que se graban las pistas y sectores mediante señales magnéticas recibe el nombre de inicialización del *diskette*.

Obviamente, un disco nuevo que no ha sido previamente inicializado por el sistema operativo (DOS) no sirve para guardar ningún tipo de información útil para el ordenador. No es el objetivo de este artículo tratar los comandos de inicialización, que por otra parte vienen explicados en el manual de la unidad. Conviene recordar que los sectores se numeran a partir del cero, así el primer sector es el referido como cero.

La pista 18 es particularmente interesante, pues se la reserva el sistema





```
TRACK 18  SECTOR 1

00 :12 04 82 11 00 48 4F 57 20 54 4F 20 55 53 45 A0 :  II HOW TO USE
10 :A0 A0 A0 A0 A0 00 00 00 00 00 00 00 00 00 10 00 :
20 :00 00 82 11 07 48 4F 57 20 50 41 52 54 20 54 57 :  II HOW PART TW
30 :4F A0 A0 A0 A0 00 00 00 00 00 00 00 00 00 07 00 :O
40 :00 00 82 13 00 44 4F 53 20 57 45 44 47 45 A0 A0 :  II DOS WEDGE
50 :A0 A0 A0 A0 A0 00 00 00 00 00 00 00 00 00 01 00 :
60 :00 00 82 13 01 44 4F 53 20 35 2E 31 A0 A0 A0 A0 :  II DOS 5.1
70 :A0 A0 A0 A0 A0 00 00 00 00 00 00 00 00 00 04 00 :
80 :00 00 82 13 03 43 4F 50 59 2F 41 4C 4C A0 A0 A0 :  II COPY/ALL
90 :A0 A0 A0 A0 A0 00 00 00 00 00 00 00 00 00 0B 00 :
A0 :00 00 82 13 09 50 52 49 4E 54 45 52 20 54 45 53 :  II PRINTER TES
B0 :54 A0 A0 A0 A0 00 00 00 00 00 00 00 00 00 09 00 :T
C0 :00 00 82 10 00 44 49 53 4B 20 41 44 44 52 20 43 :  II DISK ADDR C
D0 :48 41 4E 47 45 00 00 00 00 00 00 00 00 00 04 00 :HANGE
E0 :00 00 82 10 01 44 49 52 A0 A0 A0 A0 A0 A0 A0 A0 :  II DIR
F0 :A0 A0 A0 A0 A0 00 00 00 00 00 00 00 00 00 04 00 :
```

Figura 4. Volcado en impresora de la pista 18/sector 1.

operativo para hacer sus anotaciones. Como vimos, tal pista dispone de 19 sectores, numerados de 0 a 18. Aclaremos que todos los sectores pueden almacenar hasta 256 bytes de información, aparte de los datos referentes al número de pista, sector y otros.

El sector 0 de la pista 18 almacena el BAM, *Blocks Availability Map* (mapa de disponibilidad de bloques o sectores, como gusta llamar **Commodore**). Es utilizado por el sistema operativo para conocer en cada momento cuál es el estado de ocupación de los sectores en cada pista.

Cuando compramos la unidad de *diskette* vemos que le acompaña un disco para demostración de su funcionamiento que recibe el nombre de *Test/Demo.* Si listamos el directorio (LOAD«$», 8 y después LIST) observamos que aparece un programa titulado *«Display T & S»* o lo que es igual: Visualización de pista y sector (Track y Sector). Carguémoslo en el ordenador. En la pantalla aparece un mensaje que nos pregunta si queremos que la visualización se haga en la pantalla (*Screen*) o la impresora (*Printer*). Una vez introducido el *diskette* sonará durante unos instantes, encendiéndose también el LED frontal que indica la utilización de la unidad, y aparece en la pantalla otro mensaje: *Track, Sector?* Tecleamos 18,0 y en la impresora aparecerá lo mismo que en la figura 3. Acabamos de visualizar el contenido del BAM en forma hexadecimal de los 256 bytes que almacena dicho sector. Cada fila corresponde a 16 bytes. Inmediatamente a la derecha, tras los dos puntos, tenemos el carácter equivalente en código ASCII, muy útil para detectar las cadenas alfanuméricas contenidas en el sector. Los dos primeros bytes de cualquier sector contienen respectivamente la pista y sector del siguiente bloque (sector) al que debe ir a leer el sistema operativo. En este caso es la pista 18 (12 en notación hexadecimal) y el sector 1 ($01, siendo $ el símbolo empleado para los números hexadecimales). El tercer byte es 65 ($41) que indica que estamos trabajando con el formato empleado por la unidad **1541.** El cuarto byte no se utiliza ahora y vale $00. Entre el quinto y el ciento cuarenta y dos está contenido el BAM y cada byte está relacionado con un sector. Los bytes 144 a 159 ($90 a $9F) son 16 bytes reservados para guardar el nombre con el que bautizamos al *diskette* cuando lo formateamos. Si el nombre que le damos es más largo de este número de bytes, el sistema operativo lo truncará. Seguidamente, los bytes 162 y 163 ($A2 y $A3) guardan el número del ID, número opcional identificador del *diskette.*

Una vez concluida la transferencia de los bytes de la pista 18/sector 0 a la impresora, la pantalla nos pregunta sobre si queremos examinar algo más. Tecleando 18,1 volcaremos sobre el papel el contenido del siguiente sector, que contiene la primera parte del directorio. Igual que ocurría con el sector anterior, este emplea sus dos primeros bytes para contener la información referente al siguiente sector al que se dirigirá al sistema operativo cuando termine con él. En el ejemplo de la figura 4 se muestra que el siguiente sector sería el 4 ($04) de la pista 18 ($12). En el sector a que nos es-

```
  Byte numero                    Informacion contenida
  -------------                  ----------------------
Decimal   Hexadecimal
==========================================================
00 y 01    $00 y $01    ........Numeros de pista y sector
                                siguiente sector del di-
                                rectorio
02 a 31    $02 a $1F    ........Cabecero del 1er. fichero
34 a 63    $22 a $3F    ........Cabecero del 2o. fichero
66 a 95    $42 a $5F    ........Cabecero del 3er. fichero
98 a 127   $62 a $7F    ........Cabecero del 4o. fichero
130 a 159  $82 a $9F    ........Cabecero del 5o. fichero
162 a 191  $A2 a $BF    ........Cabecero del 6o. fichero
194 a 223  $C2 a $DF    ........Cabecero del 7o. fichero
226 a 255  $E2 a $FF    ........Cabecero del 8o. fichero

        TRACK 18 SECTOR 1
```

Figura 5. Información contenida en un sector del directorio.

Byte numero:		Informacion contenida:
Decimal	Hexadecimal	
00	$00	........Tipo de fichero, de la forma $8X, siendo X: 0 borrado, 1 secuencial, 2 programa, 3 usuario, 4 relativo
01 y 02	$01 y $02	........Pista y sector de los primeros 256 bytes del fichero
03 a 18	$03 a $12	........Nombre del fichero
19 y 20	$13 y $14	........Pista y sector del primer sector de datos para el DOS si el fichero es relativo
21	$15	........Longitud del registro
22 a 25	$16 a $19	........Sin utilizacion
26 y 27	$1A y $1B	........Pista y sector de un fichero nuevo
28 y 29	$1C y $1D	........Numero de sectores utilizados por el fichero (byte de menor peso-byte de mayor peso).

Figura 6. Estructura de cada cabecero del directorio.

tamos refiriendo vemos el nombre de 8 ficheros, que pueden ser secuenciales, relativos, programas, etc., este es el número máximo de ficheros a que se puede hacer referencia en un sector del directorio. El cuadro de la figura 5 nos ayudará a ver más claro cómo está distribuida la información que afecta a cada fichero.

Los bytes 2, 34, 66, 98, 130, 162, 194 y 226 dan comienzo cada uno de ellos al cabecero de un fichero, teniendo todos el mismo formato, según se ve en la figura 6. El byte 0 correspondiente al primer fichero de que se trata. Sea cual sea el tipo, siempre va precedido por $8, es decir, se suman 128 (decimal) al número de identificación del tipo de fichero. Así por ejemplo, 122 ($82) indica que se trata de un programa.

Los dos siguientes bytes, 1 y 2, dicen cuáles son el número de pista y sector del bloque de datos que contiene los primeros 256 bytes del fichero en sí. En realidad son 254 bytes, pues los 2 primeros bytes de ese sector son a su vez el número de pista y el de sector en que se debe buscar el siguiente bloque de 256 bytes de datos y así sucesivamente.

Los bytes que ocupan los lugares cuarto a decimonoveno contienen al nombre con que ha sido bautizado el fichero, que puede tener una longitud máxima de 16 bytes (caracteres).

El siguiente par de bytes contiene información del número de pista y sector donde se guarda una información muy útil para el sistema operativo cuando trabaja con ficheros relativos, pero que no vamos a entrar en discutir, por no ser este el objetivo del artículo.

El byte 21 almacena la longitud del fichero en el caso de que este sea relativo. Los 22 a 25 carecen de utilización y los 26 y 27 contienen una información que solamente interesa al sistema operativo.

Los bytes 28 y 29 contienen información redundante, que dice al sistema operativo cual es la longitud total del fichero en número de sectores utilizados por él. Sin embargo, como todos los sectores que albergan a un fichero están unidos por medio de los 2 primeros bytes de cada uno de ellos, cuando ya no hay más sectores que leer estos dos bytes lo indican en el último sector.

Esta misma estructura es utilizada por los cabeceros de los otros siete ficheros referidos en este sector del directorio, mantiendose el mismo orden de situación de los bytes que portan la información relevante que identifica a cada fichero.

Regresión lineal

Múltiples son las aplicaciones prácticas que permiten la utilización de la Regresión Líneal cuando disponemos de datos de naturaleza estadística o experimental. Con este programa podemos hacer los cálculos adecuados con mayor comodidad, eliminando posibles errores durante el manejo de los datos.

Como punto de partida disponemos de un conjunto de cifras (datos). Cada uno viene representado por un punto en un plano, cuyas coordenadas son (x, y). Nuestra intención es ajustar, el conjunto de puntos a una recta que tiene por ecuación genérica:

$y = ax + b$

La intención última es que la recta se ajuste lo más posible a los puntos.

El método utilizado para hallar los parámetros a y b es el llamado ajuste por «mínimos cuadrados». Los criterios utilizados en este método son:

1. Los valores de las abcisas —coordenadas en el eje x— son exactos.
2. La diferencia entre los valores introducidos «y» y el valor que tendría la curva «ax + b» —para una misma x— sea mínima.

Partiendo de estas premisas llegamos, después de una serie de operaciones matemáticas, a un sistema de dos ecuaciones con dos incógnitas:

$$\sum_{i}^{N} Y_i - N\cdot b - a\sum_{i}^{N} X_i = O$$

$$\sum_{i}^{N} X_i\cdot Y_i - b\sum_{i}^{N} X_i - a\sum_{i}^{N} X_i^2 = O$$

donde «a» representa la pendiente de la recta, «b» la ordenada, «N» el número de puntos e «y» los valores de las coordenadas de los puntos.

FUNCIONAMIENTO DEL PROGRAMA

Lo primero que hace el ordenador es pedirnos el número de puntos que vamos a introducir. Posteriormente se irán tecleando los pares de valores x, y, no tardando en aparecer en la pantalla los valores «a», «b» y «r», que son la pendiente, la ordenada y el coeficiente de correlación. Este último es un parámetro que nos ofrece una idea sobre cuánto se ajusta la recta a los puntos. Si r = 1 significa que la recta pasa por todos los puntos. Cuanto más por debajo de este valor se sitúe más dispersos se encontrarán los puntos en torno a la recta.

La pantalla también indica una serie de funciones:

G. Representación gráfica. Consiste en una visualización gráfica de los puntos y la recta. Una vez que están situados los puntos y los ejes de coordenadas, pulsando Return aparece la recta. Presionando nuevamente la tecla Return volvemos a la parte principal del programa.

E. Extrapola valores. Esto consiste en hallar valores de la recta que deseamos conocer. Se introducen los valores mínimo y máximo —correspondientes al eje x— y el intervalo con el que el valor mínimo se va acercando al máximo. Seguidamente obtendremos los valores de las «x» con sus correspondientes «y». Pulsando otra vez Return aparecerá la representación gráfica, tomando como límites los valores mínimo y máximo. A veces resulta útil esta función para comprobar dónde corta la recta a los ejes coordenados, puesto que no siempre quedan formateados dentro de la pantalla. En estos casos es aconsejable extrapolar para valores más negativos —si los puntos que hemos introducido son positivos— o viceversa.

Junto a los ejes de coordenadas aparecen unas ligeras marcas que corresponden al valor de la unidad.

I. Imágenes. Al introducir un valor x proporciona el correspondiente a f(x).

A. Antiimágenes. Introduciendo un valor para y devuelve el correspondiente a (f^{-1}(y)).

P. Parámetros. Vuelve a escribir los valores de los parámetros a, b y r.

DESCRIPCION DEL PROGRAMA

Línea 10. Define una función FNC que redondea hasta las milésimas.

Líneas 50 a 70. Introduce los valores de los puntos.

Líneas 80 a 140. Calcula los sumatorios y los coloca como coeficientes de las ecuaciones.

Línea 145. Calcula el coeficiente de correlación.

Líneas 150 a 245. Calcula la pendiente y la ordenada, resuelve el sistema de ecuaciones y los imprime en pantalla.

Líneas 250 a 253. Imprime las funciones que puede realizar el programa.

Líneas 254 a 268. Escoge la función que va a realizar.

Línea 270. Ejecuta la extrapolación de valores e imprime los resultados.

Líneas 315 a 357. Ejecuta la extrapolación de valores e imprime los resultados.

Líneas 358 a 359. Representación gráfica de los valores extrapolados entre los puntos máximo y mínimo.

Líneas 400 a 460. Encuentra los valores máximo y mínimo de entre todos los puntos introducidos y los coloca en las variables MX, NX, MY y NY.

Líneas 520 a 530. En la primera parte comprueba si los ejes de coordenadas entran en el formato de la

Programas

pantalla y en caso afirmativo salta a otra subrutina que los representa. En la última parte se sitúan los puntos.

Línea 700. Dibuja el eje de coordenadas.

Línea 710 y 711. Dibuja la posición del valor «1» en el eje de ordenadas.

Línea 720. Dibuja el eje de abcisas.

Línea 730 y 731. Dibuja la posición del valor «1» en el eje de abcisas.

Línea 740. Calcula los parámetros PX y PY que se utilizan en las rutinas de los gráficos, para que en todo momento los valores a representar queden formateados debidamente.

Líneas 750 y 770. Sitúan los puntos de la recta en la pantalla gráfica.

Líneas 780 a 810. Calcula e imprime la imagen de un número.

Línea 820 a 850. Lo mismo que en el caso anterior, pero con la antiimagen.

Joan HOMS

```
5 REM AJUSTE DE UNA FUNCION PRO EL METODO DEMINIMOS CUADRADOS
10 DEF FNC(X)=INT(X*10000+.5)/10000
15 REM CAS LINEAL AX+B
20 PRINT"■■■":POKE36879,8
22 PRINT"■***INSTRUCCIONES*":PRINT"■■■PULSA ■RETURN■ PARA     CONTINUAR ■"
30 INPUT"**NUMERO DE VALORES";N
40 DIM A(N-1,1),B(2,3)
50 FORT=0TON-1
60 INPUT"*VALOR X,Y";A(T,0),A(T,1)
70 NEXT
80 FORT=0TON-1
90 B(1,3)=B(1,3)+A(T,0)*A(T,1)
100 B(2,3)=B(2,3)+A(T,1)
110 B(1,1)=B(1,1)+A(T,0)*A(T,0)
120 B(1,2)=B(1,2)+A(T,0)
125 B(0,1)=B(0,1)+A(T,1)*A(T,1)
130 NEXT
140 B(2,1)=B(1,2):B(2,2)=N:M=B(1,2)
145 R=(N*B(1,3)-B(2,3)*B(2,1))/SQR((N*B(1,1)-B(1,2)*B(1,2))*(N*B(0,1)-B(2,3)*B(2
,3)))
150 REM RESOLUCIO DE        L'EQUACIO
160 IF SGN(B(1,2))=SGN(B(2,2))THEN N=-N
170 FORT=1TO3
180 B(1,T)=B(1,T)*N
190 B(2,T)=B(2,T)*M
200 NEXT
210 X=(B(1,3)+B(2,3))/(B(1,1)+B(2,1))
220 Y=(B(2,3)-B(2,1)*X)/B(2,2)
222 N=ABS(N)
230 PRINT"■■■**PENDIENTE A=";X
240 PRINT"■**ORDENADA B=";Y
245 PRINT"■*COEFICIENTE DE          CORRELACIO R=";R
250 PRINT"■■*REPRE. GRAFICA ■G":PRINT"■*IMAGEN  DE.. ■I"
252 PRINT"■*ANTIIMATGEN DE..  ■A":PRINT"■*EXTRAPOLAR VALORES ■E"
253 PRINT"■*PARAMETROS CURVA ■P"
254 GETA$
255 IFA$="E"THEN320
256 IFA$="G"THEN270
257 IFA$="P"THEN230
258 IFA$="I"THEN780
259 IFA$="A"THEN820
268 GOTO254
270 GOSUB400
280 GOSUB500
290 GETA$:IFA$<>CHR$(13)THEN290
300 GOSUB750
310 GETA$:IFA$<>CHR$(13)THEN310
315 FOR4:GOTO250
```

N. de R.: Aunque el autor ha desarrollado este programa utilizando las leyendas en catalán, no plantea ninguna dificultad alterar las sentencias que las llevan.
Vic 20 con Superexpander.

```
320 PRINT"■■**EXTRAPOLAR VALORS  "
325 PRINT"■**VALOR MINIM I MAXIM":INPUTNX,MX
330 IFNX>MXTHENPRINT"T'HAS EQUIVOCAT,PRIMER ENTRA EL MINIM":GOTO325
335 IFNX=MXTHENPRINT"NO S'ACEPTEN NOMBRES IGUALS":GOTO325
340 INPUT"■*INCREMENT";I:IFI<=0THENPRINT"VALOR ERRONI":GOTO340
345 PRINT"■ABCISES   ■ ■ ORDENADES":PRINT"          ■ ■":PRINT"■
    ■
348 MY=NX*X+Y:NY=MY
350 FORT=NXTOMXSTEPI
352 IFT*X+Y>MYTHENMY=T*X+Y
353 IFT*X+Y<NYTHENNY=T*X+Y
354 PRINT FNC(T),FNC(T*X+Y)
355 NEXT
357 GETA$:IFA$<>CHR$(13)THEN357
358 GOSUB740
359 GOTO280
400 REM MAXIM I MINIM
405 MX=A(1,0):NX=MX:MY=A(1,1):NY=MY
410 FORT=0TON-1
420 IFA(T,0)>MXTHENMX=A(T,0)
430 IFA(T,0)<NXTHENNX=A(T,0)
440 IFA(T,1)>MYTHENMY=A(T,1)
450 IFA(T,1)<NYTHENNY=A(T,1)
460 NEXT:GOSUB740
470 RETURN
500 REM REPRESENTACIO          GRAFICA
510 FOR2:DIM0,0,1,1
520 REM DIBUIXA EIXOS
530 Q=-PX*NX:W=10:K=W:E=PY*MY
540 IFNX<=0THENGOSUB700
550 IFNX<=0ANDMY-1>0THENGOSUB710
560 IFMY>=0THENGOSUB720
570 IFMY>=0AND1-NX>0THENGOSUB730
590 REM SITUA PUNTS
600 FORT=0TON-1
605 XX=PX*(A(T,0)-NX):YY=PY*(MY-A(T,1))
607 IFXX<0ORYY<0THEN620
610 READ2,XX,YY
620 NEXT
630 RETURN
700 INPUT#2,Q,0TOQ,1023:RETURN
710 IFQ-W<0THENW=0
711 INPUT#2,Q-W,PY*(MY-1)TOQ+K,PY*(MY-1):RETURN
720 INPUT#2,0,ETO1023,E:RETURN
730 IFE-W<0THENW=0
731 INPUT#2,PX*(1-NX),E-WTOPX*(1-NX),E+K:RETURN
740 PX=1022/(MX-NX):PY=1022/(MY-NY):RETURN
750 FORT=NXTOMXSTEP(MX-NX)/200
755 XX=PX*(T-NX):YY=PY*(MY-T*X-Y)
756 IFXX<0ORYY<0THEN770
760 READ2,XX,YY
770 NEXT:RETURN
780 REM IMATGE
790 PRINT"■■":INPUT"*VALOR PER X";A
800 PRINT"IMATGE"Y+A*X
810 GOTO254
820 REM ANTIIMATGE
830 PRINT"■■":INPUT"*VALOR PER Y";A
840 PRINT"ANTIIMATGE"(A-Y)/X
850 GOTO254
```

PROGRAMA COMPROBADO commodore Magazine

Forth para el C-64

Escribamos en el teclado *1* y pulsemos RETURN. FORTH nos responderá con el mensaje OK. Si escribimos *2 3* (y, como siempre aunque no lo volvamos a indicar, pulsamos RETURN), obtenemos el mismo resultado. Si ahora escribimos ., el ordenador nos contesta con un 3 (y el OK que nos indica que todo ha ido bien). Un nuevo . provocará la impresión de un 2 y un tercer . la del 1 introducido en primer lugar. Si insistimos y volvemos a escribir ., obtendremos un cierto número,pero seguido esta vez no de OK,sino de un mensaje de error (EMPTY STACK) que nos indica que la pila está vacía. Para entender lo que está sucediendo, debemos recordar el concepto de pila.

La pila es una de las estructuras de datos más importantes en Informática. Los microprocesadores, incluido el 6510 del C-64, usan una pila para guardar temporalmente datos y, sobre todo, para asegurar la lógica de las llamadas subrutinas y el feliz retorno desde las mismas. Los compiladores y los programas recursivos usan pilas y lo mismo hacen los intérpretes BASIC (citemos, por ejemplo, la pila que almacena con cada GOSUB la dirección a la que debe volverse con el correspondiente RETURN). Pero mientras que en la mayor parte de los lenguajes la pila (o pilas según los casos) permanece invisible al usuario, en FORTH es directamente accesible y constituye uno de los elementos centrales de la programación. Para ser más precisos, digamos que nos estamos refiriendo a la llamada «pila de datos» que vamos a pasar a describir, ya que en FORTH existe una segunda pila tan importante como aquélla, pero sobre la que no diremos nada debido a la limitación del espacio disponible.

Como su mismo nombre indica, una pila no es más que una zona de la memoria viva en la que se van apilando, es decir, almacenando unos encima de otros, números binarios (de 16 bits en el caso de FORTH) conservándose rastro únicamente de la disección en que se han guardado el primero y el último de ellos. Cada nuevo número introducido debe ser apilado (colocado) sobre el anterior y no puede ser intercalado entre elementos preexistentes en la pila. Análogamente, el único elemento de una pila que resulta ser directamente accesible es (normalmente, ya que como veremos FORTH ofrece muchas facilidades en este sentido) el que se encuentra en la cima de la pila, de modo que el primer elemento que puede retirarse de una pila es, precisamente, el último introducido. Una analogía que puede resultar sugerente es la de una pila de platos en la cual no está permitido más que colocar un plato encima de todos los demás o retirar el plato que se encuentra en dicha posición dominante y que ha sido el último introducido. En este sentido, en inglés se habla de pilas LIFO,ya que «last in, first out» significa que el último en entrar es el primero en salir.

Cuanto se inicializa FORTH la pila está vacía (al igual que lo está tras cualquier error o la ejecución de la plabra *SP!,* cuya acción consiste precisamente en vaciar la pila). Si al analizar una línea de pantalla el intérprete externo de FORTH encuentra un número, lo deposita en la cima de la pila. Así, en el ejemplo que hemos visto antes, los números 1, 2 y 3 han ido siendo colocados en !a pila por este orden. La situación de la pila en este caso se expresa simbólicamente como (1,2,3) siendo el elemento más a la derecha el que se encuentra encima de todos los demás.

La palabra . sirve para retirar el último elemento de la pila, que en este caso era el 3, e imprimirlo en pantalla. La acción de . sobre la pila puede por tanto representarse como (1,2,3 - 1,2) en este caso concreto y en general como (n -), donde el guión sirve para separar los últimos elementos de la pila antes y después de la ejecución de la palabra. También resulta ahora fácil entender que al volver a escribir . vayamos obteniendo sucesivamente 2, 1 y un número sin sentido seguido de un mensaje de error para indicarnos que la pila estaba vacía.

Por supuesto que . no es la única palabra que actúa sobre los elementos de la pila. De hecho, la mayor parte de las palabras que se encuentran en el diccionario de base y de aquellas definidas por el usuario tiene alguna influencia sobre la pila. Consideremos, por ejemplo, la palabra *DUP,* cuya acción consiste en duplicar el número que se encuentra en la cima de la pila; es decir, la misma viene simbolizada por (n - n,n). Para comprobarlo, escribamos (¡recordando que los espacios son importantísimos en FORTH!) la frase *1 2 DUP* ..., y obtendremos, tras RETURN, 2 2 1 OK. En consecuencia, la palabra *DUP* no hace sino manipular la pila, a diferencia de . que, además, imprime un número.

FORTH ofrece muchas facilidades para controlar la pila. Además de las ya mencionadas *DUP* y *SP!* podemos citar unas cuantas de las palabras más usuales cuya acción se limita a la manipulación de la pila. Por ejemplo:

- *DROP* (n -), elimina el número de la cima de la pila.
- *SWAP* (n,m - m,n), intercambia los dos últimos elementos.
- *OVER* (n,m - n,m,n), duplica en la cima de la pila el penúltimo de sus elementos.
- *ROT* (n,m,r - m,r,n), rota los tres últimos elementos.

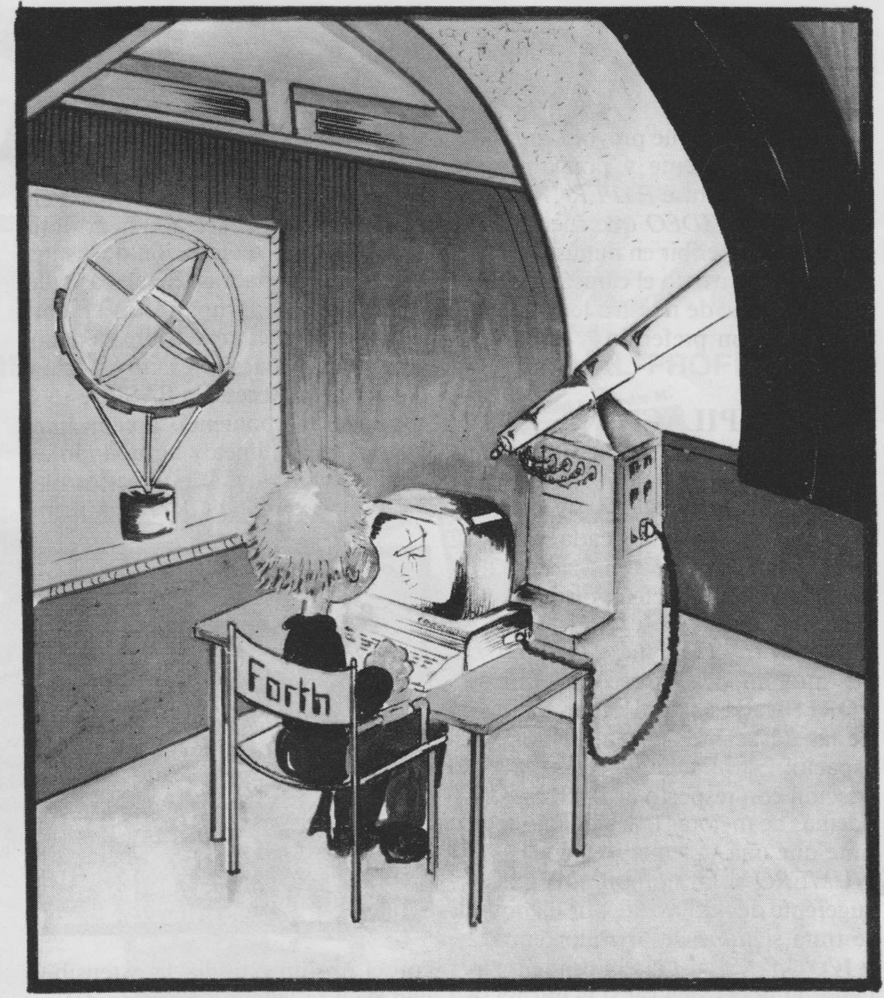
La palabra *PICK* (que por cierto no aparece en la documentación que acompaña a este FORTH a pesar de encontrarse presente en el mismo) merece una explicación más detallada, ya que permite acceder a cualquier punto de la pila. Su acción consiste en retirar el último elemento de la misma, digamos el número n, y luego duplicar el elemento que ocupa el n-simo lugar a partir del que se halla en la

(SEGUNDA PARTE)

cima de la pila tras haber retirado n. Si, por ejemplo, teníamos (5,1,9,3) en la pila y ejecutamos *PICK,* primero se retira el 3 dejándola como (5,1,9) y luego se duplica el tercer número, es decir, el 5, para dar finalmente (5,1,9,5). A estas alturas debería ser claro que 2 *PICK* tiene exactamente el mismo significado que *OVER* y que 1 *PICK* es equivalente a *DUP*.

Pero la pila no sirve sólo para guardar en ella números que pueden ser manipulados para acabar siendo impresos por medio de . . Tiene, de hecho, dos papeles centrales. El primero consiste en pasar argumentos entre palabras (en incluso entre FORTH y ENSAMBLADOR) y en este sentido juega, parcialmente, el mismo papel que las variables, incluyendo aquí a las variables locales de otros lenguajes, aunque como veremos, FORTH también dispone de variables entendidas en un sentido más habitual. Este importante papel de la pila será analizado más adelante.

La pila es, también, el lugar donde se realizan las operaciones aritméticas. En efecto, FORTH sabe realizar ciertos cálculos matemáticos, +, -, * y / entre otras, pero únicamente sobre elementos que se encuentran entre los últimos de la pila. Por ejemplo, + retira los dos últimos elementos de la misma y deposita allí su suma, es decir, su acción viene representada por (n,m - r), siendo r = n + m. Por tanto, si queremos saber la suma de 2 y 3, basta escribir la frase *2 3 +* . (equivalente al PRINT 2 + 3 del BASIC), para que FORTH nos conteste, no lo adivinaréis nunca, con un sapientísimo 5. Notemos que el orden en que van escribiéndose las cosas, conocido con el nombre un tanto pedante de *notación polaca inversa* por razones que no hacen al cuento, es muy distinto al del BASIC, aunque debería resultar más que familiar para los usuarios de una de las más conocidas marcas de calculadoras de bolsillo. Primero deben introducirse en la pila



los *operandos* (el 2 y el 3 en este caso), luego el *operador (+)* que actúa únicamente sobre números preexistentes en la pila; finalmente puede añadirse un . si se desea conocer el resultado que ha quedado en la cima de la pila.

Otro ejemplo nos viene proporcionado por la sucesión de palabras *1-* cuya acción (n - n-1) de decrementar en una unidad el último elemento de la pila es tan útil y se realiza tan repentinamente que FORTH dispone de una palabra específica para realizar esta tarea en forma más rápida y ocupando menos memoria en la compilación; se trata, bastante lógicamente, de la palabra *1-*. Existen también palabras análogas *1+, 2+ y 2-* cuya acción respectiva es (n - n+1), (n - n+2) y (n - n-2).

Existen muchas más palabras que permiten controlar la pila o realizar operaciones matemáticas en la misma, pero nos limitaremos aquí a señalar que los operadores de comparación > y < tiene como acción (n,m - f), siendo f cierto (es decir, distinto de cero) si n era mayor (respectivamente menor) que m y falso (es decir, nulo) en caso contrario. Existe una versión abreviada Ø< de la frase Ø< .

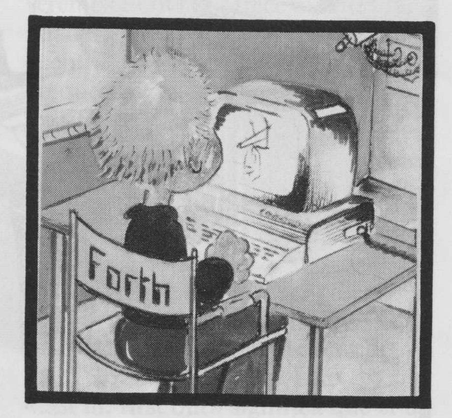
Tampoco . y *TYPE* son las únicas palabras relacionadas con la impresión. Un carácter de código ASCII n puede imprimirse por medio de *n EMIT.* Igualmente, una cadena de caracteres se imprime mediante el uso de la palabra ." en la forma: ." *MENSAJE"* (¡nótese el espacio obligatorio tras la palabra ."!) envía a la pantalla la cadena MENSAJE. Para que la impresión se produzca no en la pantalla sino en la impresora, debe usarse *PAPER* y, para volver a la pantalla, *VIDEO* (¡bastante más fácil que en BASIC!). Aunque aún hay muchas otras palabras en este apartado, citemos

únicamente CR que provoca un salto a la línea siguiente y pongamos el ejemplo de la frase *PAPER 17 EMIT ." FORTH" VIDEO* que fuerza a la impresora a escribir en minúsculas (ya que hemos enviado el carácter ASCII 17) el nombre de nuestro lenguaje de programación preferido.

LA COMPILACION

Hasta este momento, cada vez que hemos pulsado la tecla RETURN, FORTH ha analizado cada palabra escrita, entendiendo como palabra toda sucesión de caracteres distintos del espacio blanco. Así en la frase *23 4 DUP* hay tres palabras: *23, 4 y DUP.* ¡Es muy importante recordar que en FORTH cada palabra debe separarse de las adyacentes por, al menos, un espacio! Esto puede parecer una restricción con respecto al BASIC, pero además de mejorar la legibilidad, permite que una variable se llame *2-1Ø-NUMERO* si tal nombre nos parece sugerente debido a que, por ejemplo, se trata siempre de un valor entre 2 y 1Ø. Una vez aislada la primera palabra, FORTH va a ver si la misma se encuentra en el diccionario (en realidad en el vocabulario de contexto o en los vocabularios padres del mismo, pero olvidémonos de estas interesantes sutilezas que no podemos comentar aquí. Si la respuesta es afirmativa, la palabra es ejecutada y en caso contrario se intenta traducir en un número (en la base corriente que puede ser cambiada a voluntad). Si la palabra resulta ser un número, es depositado en la pila y si no lo es el proceso se detiene con un mensaje de error. FORTH continúa la interpretación de la línea introducida hasta el fin de la misma, para acabar emitiendo el mensaje OK si no se ha producido ningún error por el camino.

Esta respuesta inmediata de FORTH es común a los lenguajes interpretados (corresponde al modo comando directo del BASIC) y resulta un medio interactivo particularmente cómodo y eficaz para la puesta a punto de los programas, sobre todo cuando se tiene la modularidad de FORTH a nuestra disposición. Pero por supuesto, es necesaria la posibilidad de diferir la ejecución de las instrucciones (palabras en el caso del FORTH) y de agrupar éstas en conjuntos que puedan ser almacenados (en disco, en nuestro caso) y ejecutados repetidas veces. En BASIC esto se consigue anteponiendo a cada línea un número de línea y agrupando éstas en programas. Aunque estos últimos no existen en FORTH, al menos



con el mismo sentido, la extensibilidad de este lenguaje permite definir nuevas palabras tan complejas como se quieran y añadirlas al diccionario.

El primer método para hacer esto es usar la palabra *CONSTANT,* que utilizada en la forma *CONSTANT NOMBRE* retira el último elemento de la pila y define una palabra llamada *NOMBRE* que, cada vez que sea ejecutada, dejará en la pila aquel número. Por ejemplo, si hacemos 1Ø *CONSTANT DIEZ,* el efecto de la palabra *DIEZ* consistirá en dejar sobre la pila un 1Ø. Es claro que *CONSTANT* define constantes, es decir, números que son utilizados muy frecuentemente y que se han definido como tales constantes para ganar espacio en la compilación y tiempo en la ejecución. Con estos mismos objetivos, el FORTH del C-64 tiene definidos como constantes los números *Ø, 1, 2 y 3.*

Por supuesto que además de las constantes tenemos necesidad de variables. Estas se definen como palabras con ayuda de *VARIABLE,* que utilizada como *VARIABLE NOMBRE,* retira un número de la pila y define una palabra *NOMBRE* a la que asigna el valor inicial de aquel número y cuya acción consiste en depositar sobre la pila la dirección de memoria donde se encuentran almacenados los 16 bits correspondientes al mencionado valor. Es decir, si definimos *12 VARIABLE DOCE,* el ejecutar la frase *DOCE .,* veremos aparecer un número que no es 12, sino la dirección donde dicho valor ha sido almacenado. Para imprimir en pantalla el contenido de los dos octetos cuya dirección de memoria viene dada por el número de la cima de la pila, puede usarse *?,* una palabra cuya acción es (d -) e imprimir los 16 bits contenidos en la dirección d. Por tanto, *DOCE ?* dará 12. También puede depositarse sobre la pila el contenido de cualquier dirección de memoria, y en particular el valor de una variable, mediante la palabra @ cuya acción es (d - n) siendo n el número almacenado en la dirección d. Por tanto la frase @ . es equivalente a la palabra ?. Para cambiar el contenido de una dirección de memoria, y en particular el valor de una variable, puede usarse *!* cuya acción (n,d -) consiste en quitar de la pila los dos últimos elementos de la misma y guardar el penúltimo, n, en la dirección indicada por el último, d. Es decir, a la sucesión *11 DOCE ! DOCE ?* FORTH nos responderá con 11. Nótese que @ y *!* son análogos al PEEK y POKE en BASIC, o para ser más precisos y tener en cuenta que actúan sobre dos octetos al DEEK y DOKE de algunos BASIC, pero su utilización es muchísimo más habitual en FORTH, donde *!* hace en *11 DOCE !* el mismo papel que = en la asignación del BASIC: DO = 11.

Por analogía con las palabras *1+* y *1-* para incrementar y decrementar la cima de la pila, existen las palabras *1+!* y *1-!* cuya acción (d —) consiste en incrementar y decrementar, respectivamente, el contenido de la dirección

d. En consecuencia, si escribimos *DOCE 1+! DOCE ?,* la respuesta, 12, nos asegura que la variable *DOCE* ha recuparado su valor original.

El método más utilizado para definir nuevas palabras, sin embargo, consiste en el uso de las palabras : y ; en la forma: : *NOMBRE ... ;* donde ... representa una sucesión de número y palabras definidas previamente. La palabra : comienza la definición de una nueva palabra llamada *NOMBRE* y pone la intérprete externo en modo compilación de forma que cuando encuentra un número en vez de depositarlo en la pila, lo compila como hace también con toda palabra (con tal de que no sea una palabra inmediata que debe ser ejecutada en toda circunstancia) en lugar de proceder a su ejecución. El proceso termina al encontrar ;, si antes no se ha producido un error debido a la presencia de una palabra desconocida. En caso de compilación correcta, la palabra *NOMBRE* pasa a formar parte del diccionario (dentro del llamado vocabulario corriente) y a partir de este momento puede usarse en pie de igualdad con cualquier´otra palabra, siendo su acción la sucesión de aquellas correspondientes a los números y palabras incluidas entre *NOMBRE* y ;. Pongamos algunos ejemplos:

: CONSTANTE CONSTANT ;
: 2 2 * ;*
: DOBLE DUP + ;

La palabra *CONSTANTE* es la versión castellanizada de *CONSTANT* y tendrá exactamente su misma acción. Por otro lado, *2** servirá para multiplicar por 2 la cima de la pila. Así, *12 2* .* dará lugar a una respuesta de 24, ya que la evolución de la pila habrá sido (12 - 12,2 - 24 -). Finalmente, *DOBLE* tiene exactamente el mismo efecto que 2* aunque realizado en distinta forma ya que tras *12 DOBLE* la pila pasa de (12) a (12,12) para dar finalmente (24).

Es claro que esta forma de construir nuevas palabras por yuxtaposición de otras anteriores corresponde a la llamada de subrutinas o procedimientos por nombre (en lugar de por número de línea como se hace en BASIC), efectuándose el paso de parámetros a través de la pila. Es decir, cada nueva palabra definida en función de las anteriores es el equivalente de un procedimiento en lenguajes como LOGO o PASCAL y se utiliza la pila para los parámetros a pasar y las variables locales. En la cadena de llamadas a palabras (o procedimientos) de nivel inferior siempre se acaba por llegar a palabras primitivas, definidas en código máquina y directamente ejecutables.

Pero una de las cualidades de FORTH es la facilidad y naturalidad con las que el usuario puede definir nuevas primitivas. En efecto, usando, en lugar de : y ;, *CODE* y *ENDCODE* junto con vocabulario *ASSEMBLER,* pueden definirse palabras en ensamblador cuyo estatus posterior será equivalente al de las palabras definidas directamente en FORTH.

Notemos que *CONSTANT, VARIABLE, :* y *CODE* son palabras que permiten definir nuevas palabras. Pero la extensibilidad de FORTH es tan grande que pueden definirse palabras que cuando sean ejecutadas definan a su vez nuevas palabras. Esto puede hacerse en un sentido trivial como el visto anteriormente en el ejemplo de *CONSTANTE* o en uno mucho más profundo que hace uso de las palabras *<BUIDS y DOES>* junto a las enormes posibilidades de controlar y dirigir el proceso de compilación que FORTH nos ofrece. No tenemos espacio para indicar aquí cómo puede hacerse esto ya que exige un cierto conocimiento de cómo se compilan las nuevas palabras y, por tanto, nos limitaremos a citar uno de los muchos ejemplos en los cuales esta posibilidad resulta útil. El FORTH del C-64 carece de variables dimensionadas. Es sin embargo muy fácil dotarle de una palabra, que por analogía con el BASIC podemos llamar *DIM*, y que al ser ejecutada defina a su vez una variable con dimensiones al igual que *VARIABLE* define una variable habitual.

Digamos, finalmente, que esta habilidad y la capacidad de controlar completamente el proceso de compilación (inexistente en otros lenguajes) harán las delicias de todo usuario al que entusiasme la programación personalizada. Si además es aficionado el código máquina, tendrá a su disposición una variante correspondiente al lenguaje ensamblador y proporcionada por la palabra *;CODE.*

Juan María y José Miguel AGUIRREGABIRIA
Departamento de Física de la Facultad de Ciencias y Escuela Universitaria del Profesorado de EGB, Universidad del País Vasco.

Fisher-Price®

ayuda a los pequeños a descubrir el mundo de los ordenadores. Porque sabe que los adultos del futuro lo necesitan asi.

Desde hace muchos años Fisher-Price viene siendo parte del mundo de los niños, ayudando a los pequeños a desarrollar importantes habilidades mientras juegan.

Hoy en día los niños están creciendo en un mundo nuevo. Un mundo en dónde los ordenadores posibilitan nuevas y emocionantes experiencias. Esta es una razón suficiente para lanzar un nuevo producto Fisher-Price:

«Fisher-Price, Juegos educativos para ordenadores»

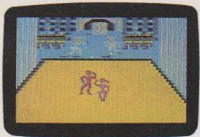


Hemos estado trabajando durante mucho tiempo con los principales diseñadores de programas, para estar seguros de que nuestros juegos ofrecen una equilibrada combinación de diversión y valor pedagógico. Nuestros programas cubren cinco áreas clave en el desarrollo del niño: aritmética, lenguaje, creatividad, habilidades generales y conocimientos básicos de microinformática.

Para los menores de 8 años disponemos de los programas con orla roja, para los de hasta 12 años disponemos de los programas con orla azul. Descubrirá que los juegos han sido cuidadosamente diseñados para la edad y nivel de desarrollo apropiados.

Nuestros programas están disponibles en castellano con versiones en cassette para Comodore-64 y Sinclair Spectrum.

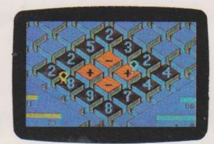
No dude en conseguir la gama completa de «Fisher-Price, Juegos Educativos para Ordenadores»... por su valor, diversión y calidad educativa.



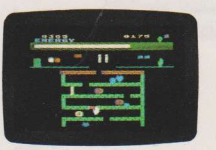
¡BAILEMOS!



SALTA Y SUMA



NUMEROS LOCOS



LABERINTOS LOGICOS

Producido y distribuido en España por: IDEALOGIC® S.A. Barcelona

